Master Thesis

Controlled Languages in Software User Documentation

by

Dina Dervišević
Henrik Steensland

LITH-IDA-EX--05/070--SE

2005-09-23
In order to facilitate comprehensibility and translation, the language used in software user documentation must be standardized. If the terminology and language rules are standardized and consistent, the time and cost of translation will be reduced. For this reason, controlled languages have been developed. Controlled languages are subsets of other languages, purposely limited by restricting the terminology and grammar that is allowed.

The purpose and goal of this thesis is to investigate how using a controlled language can improve comprehensibility and translatability of software user documentation written in English. In order to reach our goal, we have performed a case study at IFS AB. We specify a number of research questions that help satisfy some of the goals of IFS and, when generalized, fulfill the goal of this thesis.

A major result of our case study is a list of sixteen controlled language rules. Some examples of these rules are control of the maximum allowed number of words in a sentence, and control of when the author is allowed to use past participles. We have based our controlled language rules on existing controlled languages, style guides, research reports, and the opinions of technical writers at IFS.

When we applied these rules to different user documentation texts at IFS, we managed to increase the readability score for each of the texts. Also, during an assessment test of readability and translatability, the rewritten versions were chosen in 85% of the cases by experienced technical writers at IFS.

Another result of our case study is a prototype application that shows that it is possible to develop and use a software checker for helping the authors when writing documentation according to our suggested controlled language rules.
Master Thesis

Controlled Languages in Software User Documentation

by

Dina Dervišević
Henrik Steensland

LITH-IDA-EX--05/070--SE

2005-09-23

Supervisor and Examiner: Magnus Merkel
Abstract

In order to facilitate comprehensibility and translation, the language used in software user documentation must be standardized. If the terminology and language rules are standardized and consistent, the time and cost of translation will be reduced. For this reason, controlled languages have been developed. Controlled languages are subsets of other languages, purposely limited by restricting the terminology and grammar that is allowed.

The purpose and goal of this thesis is to investigate how using a controlled language can improve comprehensibility and translatability of software user documentation written in English. In order to reach our goal, we have performed a case study at IFS AB. We specify a number of research questions that help satisfy some of the goals of IFS and, when generalized, fulfill the goal of this thesis.

A major result of our case study is a list of sixteen controlled language rules. Some examples of these rules are control of the maximum allowed number of words in a sentence, and control of when the author is allowed to use past participles.

We have based our controlled language rules on existing controlled languages, style guides, research reports, and the opinions of technical writers at IFS.

When we applied these rules to different user documentation texts at IFS, we managed to increase the readability score for each of the texts. Also, during an assessment test of readability and translatability, the rewritten versions were chosen in 85% of the cases by experienced technical writers at IFS.

Another result of our case study is a prototype application that shows that it is possible to develop and use a software checker for helping the authors when writing documentation according to our suggested controlled language rules.
Writing this thesis has been a valuable experience for both of us before we finally earn our degree as Masters of Science in Information Technology at Linköpings universitet. We have been working at IFS during two semesters and it has not always been easy. Fortunately we have had many people that helped us with this thesis, and we would like to mention some of them here.

First we would like to thank our examiner and supervisor at Linköpings universitet, Magnus Merkel, for his guidance and valuable and quick feedback on our thesis work. We are also grateful to our supervisor at IFS, Pär Hammarström, who has had many interesting ideas during our thesis work.

Our case study was made possible because of the following helpful people working at IFS, who were kind enough to let us interview them: Henrik Celinder, Susanna Gabrielsson, Hans Sjöqvist, Marietta Lovendahl, and Petra Holmberg.

We are also grateful to our opponents Christian Söderström and Anna Berglund for their never-ending patience and many valuable opinions about this thesis report.

We would also like to thank Lars Steensland who has given us linguistic advice during the development of our controlled language rules.

A special thanks goes to Alexander Lowe who was kind enough to lend us a copy of Ericsson English and who shares our interest in controlled languages.

Finally, we would like to give a very special thanks to our dear friend Åsa Karlsson, who supported us and encouraged us during the work on this thesis.

Linköping, September 2005
## Contents

1 Introduction  
   1.1 Background ................................................. 1  
   1.2 Goal ............................................................ 1  
   1.3 Case study at IFS  
      1.3.1 IFS AB ................................................... 2  
      1.3.2 Research questions ................................... 3  
   1.4 Method .......................................................... 4  
      1.4.1 Literature study ........................................ 4  
      1.4.2 Interviews ................................................. 5  
      1.4.3 Alternative methods .................................... 6  
   1.5 Limitations ..................................................... 6  
   1.6 Document outline ............................................. 7  
   1.7 Target group .................................................. 7  
   1.8 Division of work .............................................. 8  

I Theoretical background  

2 Comprehensibility and Readability  
   2.1 Definition of comprehensibility  
      2.1.1 Comprehensibility during the reading process ........ 12  
   2.2 Definition of readability  
      2.2.1 Readable as Comprehensible  
   2.3 Factors that affect readability  
      2.3.1 Lexical level ............................................ 15  
      2.3.2 Syntactic level ......................................... 16  
      2.3.3 Semantic level .......................................... 17  
   2.4 Comprehensibility measures  
      2.4.1 Comprehensibility on small versus large texts ....... 22  
   2.5 Readability measures  
      2.5.1 The assessment test .................................... 23
## 3 Translatability

3.1 The goal of translation ........................................... 27
3.2 The translator’s comprehension ................................. 28
3.3 Target language independence ................................. 29
3.4 Semantic or communicative translation ......................... 29
3.5 Common translation problems ................................. 30
  3.5.1 Lexical ambiguity ........................................... 30
  3.5.2 Syntactic ambiguity ........................................... 31
  3.5.3 Contextual ambiguity ........................................... 32
  3.5.4 Anaphora .................................................... 33
3.6 Machine translation ........................................... 34
  3.6.1 Adapting to machine translation .......................... 35
  3.6.2 Typical machine translation problems ...................... 35
3.7 Translation memories ........................................... 36
3.8 Parameters of correspondence ..................................... 37
3.9 Translatability measures ........................................... 38
  3.9.1 Study 1: Measuring general translatability ................ 38
  3.9.2 Study 2: Evaluating machine translatability ............... 40
  3.9.3 The BLEU method ........................................... 41

## 4 Style guides for writing

4.1 Definition of a style guide ........................................... 45
4.2 Benefits of a style guide ........................................... 46
4.3 Well-known style guides ........................................... 46
  4.3.1 Microsoft Manual of Style for Technical Publication .......... 46
  4.3.2 The Chicago Manual of Style ..................................... 46
4.4 Plain Language .................................................... 47
4.5 Style for technical writing ........................................... 49
  4.5.1 Instructions .................................................... 49
  4.5.2 Style for descriptive and explanatory writing ............... 50
  4.5.3 Sequences and warnings ........................................... 51
  4.5.4 Voice ........................................................ 51
  4.5.5 Tense and mood ................................................... 52
  4.5.6 Contractions ..................................................... 52
  4.5.7 Text functions .................................................... 53
  4.5.8 Syntactic cues ................................................. 53
##CONTENTS

###II Controlled languages

####5 Introduction to controlled languages

- 5.1 Controlled language components ........................................ 58
- 5.2 Sublanguages ................................................................ 58
- 5.3 The first controlled language ............................................. 59
- 5.4 Comprehensibility and readability of controlled languages .... 59
- 5.5 Translatability of controlled languages .............................. 60
- 5.6 Human-oriented and machine-oriented CL .......................... 61
- 5.7 Drawbacks of CLs ............................................................ 62

####6 Terminology management

- 6.1 Terminology definitions ...................................................... 65
- 6.2 Functions of terminology .................................................. 66
- 6.3 One concept, one term ...................................................... 67
- 6.4 Structuring concepts ....................................................... 68
- 6.5 Term collection .............................................................. 68
- 6.6 Term creation ................................................................. 69
- 6.7 Term classification .......................................................... 70
- 6.8 Term translation .............................................................. 71
- 6.9 Term database design ...................................................... 72
  - 6.9.1 General term fields ................................................... 72
  - 6.9.2 Fields for the translator ............................................. 73
  - 6.9.3 Grammatical fields ................................................... 74
- 6.10 Controlled language vocabularies .................................... 74

####7 Controlled language in industry

- 7.1 Caterpillar Fundamental English ....................................... 77
- 7.2 Caterpillar Technical English ............................................ 78
- 7.3 PACE ............................................................................ 78
- 7.4 AECMA Simplified English .............................................. 79
- 7.5 Ericsson English ............................................................ 80
- 7.6 Scania Swedish ................................................................ 81
- 7.7 EasyEnglish .................................................................... 81
- 7.8 Controlled language interaction ....................................... 82
- 7.9 Introducing controlled languages ..................................... 82

####8 Controlled Language Tools

- 8.1 Definition of CL checkers .................................................. 85
  - 8.1.1 Red light/green light or interactive systems .................. 86
  - 8.1.2 Prescriptive or prescriptive approach ......................... 86
## 8.2 Common problems for CL checkers

8.3 CL checkers in industry

8.3.1 PACE Checker

8.3.2 The Boeing checkers

8.3.3 Scania Checker

8.3.4 EasyEnglish

8.3.5 Maxit

8.4 Authoring memories

8.5 Measures

### III Case study

9 Documentation at IFS

9.1 IFS’ view on user assistance

9.2 IFS’ documentation structure

9.3 IFS’ document types

9.4 IFS’ documentation process

9.5 IFS Style and Writing Guide

9.6 Known problems in IFS documentation

9.6.1 Sentence length and complexity

9.6.2 Terminology

9.6.3 Other comprehensibility and translatability issues

10 IFS plans

10.1 IFS term database

10.1.1 Design

10.1.2 Extraction of terms

10.1.3 Term relations

10.2 Controlled languages at IFS

11 Rule suggestions

11.1 Approved words

11.2 Sentence length

11.3 Sentence complexity

11.4 Paragraph length

11.5 Sequences

11.6 Contractions

11.7 Compound terms

11.8 Articles

11.9 Verb forms
List of Tables

2.1 Major types of cohesive ties ........................................... 18
2.2 Thematic roles ................................................................. 20
2.3 Reading ease categories ..................................................... 24

6.1 Information fields in a term database (Sager, 1990, p.144) .......... 73
6.2 Verb-preposition associations in COBUILD .......................... 74
6.3 Example from the Ericsson English vocabulary (Ericsson, 1983) 75
6.4 Example from AECMA SE vocabulary (AECMA, 2004) .......... 75

11.1 Verb tenses ................................................................. 116

12.1 Text analysis of About Vertex Integration ............................. 130
12.2 Text analysis of Create Company In All Components ............. 130
12.3 Text analysis of Automatic Sourcing Candidates ................... 131
12.4 Result of interviews ...................................................... 131

C.1 Nouns, prepositions, and participles in the original version ...... 172
C.2 Pronouns, adverbs, and verbs in the original version .............. 173
C.3 Nouns, prepositions, and participles in the rewritten version .... 174
C.4 Pronouns, adverbs, and verbs in the rewritten version .......... 175
List of Figures

9.1 The help system at IFS ................................. 96
9.2 IFS text types ............................................ 97

10.1 Translation and User Support project at IFS ........ 103
10.2 IFS term database ....................................... 104

14.1 Our CL checker .......................................... 138
14.2 Our sample application ................................ 139
14.3 Our modified user interface ......................... 140
Chapter 1

Introduction

1.1 Background

The writing style of technical documentation has a strong affect on readability, comprehensibility, and translatability into other languages (Haller and Schütz, 2001). In order to maintain good comprehensibility and readability, and facilitate the translation work, the language rules and terminology used in the user documentation should be standardized and consistent.

Any quality improvement in the production chain of technical documentation results in multiple benefits for translation companies and reduce time and cost for translation (Haller and Schütz, 2001). This is the reason why there are efforts all over the world to use software systems with linguistic intelligence in order to help technical writers with their work (Haller and Schütz, 2001).

Controlled Language is often seen as a long sought killer application of language technology, according to Haller and Schütz (2001). Until today controlled languages have received broad attention only in the air and space industry (Haller and Schütz, 2001). This is because English is the language of communication in those areas. English is used in maintenance literature and air traffic control communication (Haller and Schütz, 2001). A controlled language consists of a set of language rules that the authors have to follow and a vocabulary of allowed words (Haller and Schütz, 2001).

1.2 Goal

The purpose and goal of this thesis is to investigate how using a controlled language can improve comprehensibility and translatability of software user documentation written in English.
1.3 Case study at IFS

In order to accomplish our goal, we have performed a case study at IFS AB\(^1\). In this section, we first give a brief overview of the company and discuss the project where our thesis fits. Then, we specify a number of research questions that will help satisfy some of the goals at IFS and, when generalized, will fulfill the goal of this thesis.

1.3.1 IFS AB

The work with this thesis was carried out at IFS in Linköping, Sweden. IFS develops and supplies component based Enterprise Resource Planning (ERP) systems for medium sized and large enterprises and organizations. IFS is a global business applications supplier, with sales in 45 countries and more than 350,000 users worldwide (IFS, 2004a). The company was founded in 1983 in Linköping, Sweden and the company’s headquarters are located in Linköping (IFS, 2004a).

R&D

During the time that we worked on our thesis at IFS, we were placed at R&D\(^2\) where a large part of the research and product development for the company is done. IFS R&D has 600 employees and is situated at offices in Sweden, Norway, USA, and Sri Lanka (Hammarström, 2004a).

IFS Applications

The ERP system that IFS offers their customers is called IFS Applications. The product offers approximately 60 application components used in manufacturing, supply chain management, customer relationship management, service provision, financials, product development, maintenance, and human resource administration (IFS, 2004a). The components are combined together to fit the IFS clients’ needs.

Since IFS Applications consist of a large number of components there exists a large amount of documentation in text form. The documentation has two main purposes—one is to be used as a reference for experienced users, and the other is to instruct and give an understanding of the functions in IFS Applications for novice users.

---

\(^1\)Industrial & Financial Systems AB; furthermore in this thesis called IFS.

\(^2\)(Department for) Research and Development
CHAPTER 1. INTRODUCTION

Project Montgomery

Our thesis fits best within Project Montgomery—IFS’ project for developing the next generation of user documentation (Hammarström, 2004a). This is now merged with the Edge project, which will result in the next core version of IFS Applications. We have been working to improve a crucial aspect of Project Montgomery, called Translation User Support.

The objectives of Translation User Support, as quoted from Hammarström (2004b), are:

1. Cost for production and translation of application and user documentation should be reduced by at least 50% in comparison to the 2004 release

2. A consistent and translatable language based on a well defined term catalog and style guide (a big step toward simplified English)

3. The documentation should be perceived as useful (on a scale from not useful, neither nor, useful and very useful) by customers and partners

4. It should be possible to produce documentation and translations in parallel with the development of the software in order to reduce time to market. (Hammarström, 2004b)

The user documentation for IFS Applications is a very large corpora consisting of approximately 3 million words. These documents need to be translated into 23 different languages. The translation work on IFS user documentation is done by an external translation agency (Hammarström, 2004a). Before a text is sent for translation, it has to go through an internal linguistic review done by experts at IFS (Hammarström, 2004a).

1.3.2 Research questions

- Do the reasons for introducing controlled languages in industry in the past match the goals of IFS; i.e., to improve comprehensibility and translatability?

- Which simple measures of comprehensibility and translatability can be used to evaluate controlled language for software user documentation at IFS?

- What are the main components of existing controlled languages and how can they be used at IFS? Are they applicable and useful?
• What are the main problems encountered in comprehensibility and translatability of software user documentation at IFS, and how can they be solved by controlled language rules?

• How is it possible to aid the writer at IFS in his work, so that his language will conform to the rules mentioned above?

1.4 Method

We have chosen to use a qualitative method for our thesis. The qualitative method in our thesis includes a literature study and a case study. In this section, we describe the procedure we used in order to find an answer to our research questions. Then, we present some alternative methods that could have been used instead.

To find the answer to our research questions, we divided the work into several steps. The first three research questions demanded a lot of reading about the work done in the area so far. The most relevant areas for our first two research questions were comprehensibility, readability, translatability, and style guides for writing. The third research question involved a detailed literature study of controlled languages in general, the existing controlled language in industry today, and existing controlled language tools. In order to answer the second part of this research question, we conducted applicability tests/interviews with several people working with documentation at IFS.

Our final two research questions are of a more practical nature. The fourth research question was answered through interviews with technical writers and other people working with the quality improvement for the user documentation at IFS. The fifth and last research question included a development of a prototype for our own style checker.

1.4.1 Literature study

We have studied a large amount of literature relevant for our area of study in order to gain the background knowledge necessary for the conduct of this study. As sources, we have primarily used books, scientific articles, proceedings, technical reports, and masters theses because of their high reliability. We have also used Internet documents written by researchers who are well known in our area of study. We have consistently used first hand sources; only in a few cases where we have been unable to locate the referenced document have we allowed ourselves to use second hand sources. In order to get a better understanding of our area of study, we have tried to find several sources to important research results presented in our thesis.
1.4.2 Interviews

In order to gather as much information as possible to facilitate our case study, we have conducted interviews with five IFS employees currently working with writing, correction and translation of documentation at IFS. We have used unstructured interviews as interview methodology because unstructured interviews usually result in greater knowledge depth. Unstructured interviews also enabled positive and negative aspects of interest to be identified in greater detail than structured or semi-structured interviewing. The downside of this methodology is that it is easy to lose focus of inquiry.

The first two persons interviewed were Susanna Gabrielsson from IFS Distribution and Hans Sjöqvist from IFS Financials. At the moment, both of them are working with terminology standardization. We chose to interview them because we wanted to study the differences in the way the documentation is written depending on the writers backgrounds. They work with the same assignments with terminology even though their backgrounds differ. Susanna Gabrielsson is a documenter and Hans Sjöqvist is a system designer.

Susanna Gabrielsson feels that it is important that the language used in IFS documentation is standardized and followed by everyone at IFS. She feels that a controlled language is a good way to achieve this. Hans Sjöqvist feels that the most important thing as a reviewer of documentation is to make sure that the information given in the documentation is correct. Stylistic aspects are not important; it is something that should be standardized and followed only by those who write documentation.

The third person interviewed was Marietta Lovendahl, a technical writer working with linguistic reviews of documentation at IFS in Sri Lanka. We interviewed Marietta Lovendahl in order to find out which problems exist within the IFS technical documentation and also to identify some language rules that would solve common comprehensibility problems and improve IFS texts.

The fourth person interviewed was Petra Holmberg who works as translation coordinator. We interviewed her in order to find out more about common translatability problems with IFS documentation.

Finally, the fifth person interviewed was Henrik Celinder, who is a former technical writer. He now works with with linguistic reviews of documentation at IFS in Sri Lanka. We interviewed him during the test of applicability of our language rules in order to find out if the rules are possible to use during the writing of documentation at IFS.

The first round of interviews was done with the the first four persons mentioned above in order to gather information about problems with IFS
documentation and to get ideas for useful controlled language rules.

The second round of interviews was done in order to test the applicability of our developed rules. These interviews were done with Henrik Celinder and Susanna Gabrielsson.

1.4.3 Alternative methods

We chose a qualitative method since it was not possible for us to interview all of IFS’ technical writers, documenters, reviewers and coordinators relevant to our study during the time we worked there. An alternative method for conducting our thesis could have been to use a quantitative method, for instance by sending out a questionnaire to translators with yes and no questions about controlled language rules and later statistically analyze the answers in order to determine which rules are useful. Another alternative quantitative method could have been to measure the time it takes to read a text written in our proposed controlled language.

1.5 Limitations

When planning the work for this thesis, many interesting aspects of controlling languages came up. In order to be able to focus on some of the areas, we have had to limit our studies of others.

This is a list of related topics not dealt with in this thesis:

Term selection In order to develop a term database, a necessary decision is which terms should be included. Although we do discuss inclusion of common words, we have not looked into the various methods of selecting individual words.

Basic errors in source text When developing our prototype application, we will assume that the input texts are spelled right and are grammatically correct. A reason for this is that we do not want to copy the behaviors of proof reading tools and grammar checkers, both of which there exist a great number with good performance.

Errors typical for non-native writers There is a possibility that non-native writers of English have a tendency to make a certain kind of mistakes more often than native writers. Some of these mistakes are simple grammatical errors, and are therefore already excluded from this thesis; however, some might be on a semantic level. It is also probable that the mother tongue of the writer has an effect on the
resulting source text. In this thesis, we will not look at this type of mistake in general; however, since we are discussing typical errors made at IFS, some of the errors that we find might be results of the writers' backgrounds.

**Presentation aspects of style guides** When discussing style guides, we will focus on the language aspects of style guides and not presentation aspects since the language aspects are more relevant to the subject of controlled language. The understanding of written text depends on three distinct components: legibility, comprehensibility, and readability. Spuida (2002). In this thesis we will not concern us with legibility.

### 1.6 Document outline

This thesis is divided into four parts following this introduction to the thesis itself.

The first part gives a theoretical background to concepts related to controlled languages. The concepts that we explain are Comprehensibility, Readability, Translatability and Style Guides. These concepts will be used later in the thesis. Readers with good linguistic knowledge might want to start at Part II and later jump back to the theoretical background if some reference is unclear.

The second part introduces the reader to the field of controlled languages. The focus in this part is on existing controlled languages and tools with respect to their purposes, designs, and achievements.

In the third part, we present our case study at IFS. In this study, we try to find answers to our research questions presented in Section 1.3.2.

Finally, in the fourth part, we discuss generalization of our findings, evaluate our work, and summarize our conclusions.

### 1.7 Target group

This thesis is intended for anyone interested in controlled languages and with some insight into the software industry. In order to understand the theoretical parts, some basic linguistic knowledge is preferred, even though we have tried to explain unfamiliar terms. The implementation chapter requires some knowledge of software development. However, there should be no problem for any interested person to understand the steps of our work.
1.8 Division of work

This thesis is a result of two authors' work and we are both responsible for the report as a whole. However, we have divided the research and writing in the following way:

- Dina Dervišević has focused on the thesis introduction, comprehensibility and readability, style guides for writing, documentation at IFS, and the final conclusions of the thesis and has had the main responsibility for chapters 1, 2, 4, 9, and 15–16. Also, she is responsible for Appendix C.

- Henrik Steensland has focused on translatability, controlled languages, terminology management, controlled language tools, IFS plans for a term database, rule suggestions, implementation of a prototype and has had the main responsibility for chapters 3, 5–8, and 10–14. Also, he is responsible for Appendix A, Appendix B, Appendix D, and Appendix E.

Through the process, we have helped each other with material and advice. We have also assisted each other in the writing, which means that the person responsible for a chapter is not the only author of it.
Part I

Theoretical background
Chapter 2

Comprehensibility and Readability

Comprehensibility is a difficult area to grasp and describe. In this chapter we try to explain and unravel some concepts and some views that exist on comprehensibility and readability. This will ease the understanding of discussions about these areas throughout the entire thesis. We also present the problem of defining and measuring comprehensibility and readability.

The software user documentation that we study exists in written form as texts. When we discuss comprehensibility we mean text comprehensibility, not comprehensibility as a general concept, which is a large area of study in cognitive science\(^1\) and psychology.

The difference between comprehension and comprehensibility is that the first concept is a mental process while the second is a property of the text. Text comprehensibility is dependent on many factors, such as the text perspective, the abstraction, the context, the complexity, and the redundancy.

Örneholm (1999), recommended by The Swedish Centre for Terminology, has a different view on comprehensibility. According to Örneholm (1999), comprehensibility is a property of the reader, not the text, and varies from person to person. He argues that when examining comprehensibility of a text, it is impossible to measure something in the text itself to be able to comment about the degree of difficulty (Örneholm, 1999).

\(^{1}\)Cognitive science is the interdisciplinary study of mind and intelligence, embracing philosophy, psychology, artificial intelligence, neuroscience, linguistics, and anthropology (Thagard, 2004).
2.1 Definition of comprehensibility

Defining comprehensibility is not an simple task, according to Nilsson (2005) at The Swedish Centre for Terminology. The concept is often connected to readability, which can be seen as a text's measurable comprehensibility; i.e., readability indices, long or short sentences (Nilsson, 2005). Readability does not say anything about the reader's experience of reading ease (Nilsson, 2005). Readability is a property of the text in the same way as length is a human property, according to Björnsson (1968). When examining text readability, we have to work with language properties which make a text more or less available to the reader (Björnsson, 1968).

Another view on comprehensibility is presented by Thüring et al. (1995), who state that the major purpose of reading a text is comprehension. In cognitive science COMPREHENSION is described as the construction of a mental model that represents the objects and semantic relations described in the text (Thüiring et al., 1995).

Over the last 30 years, there has been a large amount of research in text comprehension, mostly in the fields of psychology and education (Foltz, 1996). Text comprehension researchers have studied text comprehensibility by attempting to understand how the reader factors and the text factors influence the ease of comprehending a text (Foltz, 1996).

2.1.1 Comprehensibility during the reading process

K.S. Goodman (as cited in Kohl, 1999) described the reading process as a psycholinguistic guessing game. The reading process involves comprehension on different levels: word, phrase, clause, sentence, paragraph, and text. During the reading process, various kinds of information are involved in the comprehension process; e.g., lexical, syntactic, semantic, and thematic knowledge, as well as discourse and prosody knowledge.

The reading process is controlled by the reading goal and reading situation (Gunnarsson, 1982). She categorizes reading goals into the following categories:

1. To memorize a text surface; i.e., the syntax, the morphology, and the functional words
2. To register the contents of a text
3. To understand the sender's reality description
4. To integrate the contents of a text into the reader's own perception of reality
5. To gain direct action oriented understanding

According to Gunnarsson (1982) 
**comprehensibility depth** depends on the reader’s **reading goal**. The first two goals are the ones that most researchers refer to when talking about comprehensibility in general, according to Gunnarsson (1982). This is also the working definition of comprehensibility that we use in our thesis. Those levels deal with syntactic and semantic structure and can be quantified. Gunnarsson (1982) states that comprehensibility is deep text understanding and that it is not trivial to measure.

### 2.2 Definition of readability

Platzack (1974) and DuBay (2004) summarize the **readability** as a property of the text that makes some texts easier to read than others. There is no standardized definition of the readability concept and almost every readability researcher has his own definition (Shelby, 1992; DuBay, 2004). George Klare, one of the leading figures in readability research, gives the following definition of readability (as cited in DuBay, 2004):

> [Readability is] the ease of understanding or comprehension due to the style of writing. (DuBay, 2004, p.3)

This definition focuses on writing style as separate from issues such as content, coherence, and organization (DuBay, 2004). Gretchen Hargis at IBM gives the following definition of readability:

> [Readability is] the information’s ability to be read and understood. Readability depends on things that affect the readers’ eyes and minds. Type size, type style, and leading affect the eye. Sentence structure and length, vocabulary, and organization affect the mind. (Klare, 2000, p.129)

G. Harry McLaughlin, the creator of the SMOG reading formula, defines readability (as cited in DuBay, 2004):

> [Readability is] the degree to which a given class of people find certain reading matter compelling and comprehensible (DuBay, 2004, p.3)

This definition emphasizes the interaction between the text and a class of readers of known characteristics, such as reading skill, prior knowledge, and motivation (DuBay, 2004).

The readability concept is used in three ways, according to Gunnarsson (1982):

---

**CHAPTER 2. COMPREHENSIBILITY AND READABILITY**

---
• To indicate legibility of either handwriting or typography
• To indicate ease of reading due to either the interest-value or the pleasantness of writing
• To indicate ease of understanding or comprehension due to the style of writing

We will view the readability concept as described in the last item in Gunnarsson’s list.

2.2.1 Readable as Comprehensible

Another view on readability and comprehensibility was published in 1963 by George Klare. In his article *The Measurement of Readability* he describes readability as the ease with which material can be read but not necessarily the ease with which it can be understood (Hargis, 2000). The studies on readability at that time did not clearly show that increased readability correlated with increased comprehensibility (Hargis, 2000). However, according to Hargis (2000), a few decades later, research could show that readability is synonymous with comprehensibility.

Gunnarsson (1982) has a different view on the relation between readability and comprehensibility. Gunnarsson (1982) states that the difference between the terms *readability* and *comprehensibility* is that a text is readable when the first two reading goals are fulfilled, but the text is comprehensible when the deeper reading goals, 3–5, are achieved; see Section 2.1.1 for a further discussion about the reading goals. The psychological correspondence to the term readability is only a superficial understanding of text and is different from comprehensibility, which is a term that should be reserved for text properties of significance for deeper understanding, according to Gunnarsson (1982).

In the article *Readability Formulas—One more time* written by Shelby (1992), the following quote describes the problems with defining and describing readability:

> When discussing readability, scholars tend to agree that it is a good thing. What readability is, however, and how a writer attains readable writing is less clear. (Shelby, 1992, p.486)

According to Shelby (1992), the reason for readability being a controversial issue is that the researchers do not know what the various factors are that affect readability and the appropriate way to measure those factors.
2.3 Factors that affect readability

There is no simple answer to the question of what makes texts that we read easy or difficult, according to Platzack (1974). An interesting thing about reading ease is that it is easier to tell what makes a text difficult to read than it is to tell what makes a text easy to read (Melin, 2004).

A great number of factors on different levels of the text play an important role in making texts easier or more difficult to read and understand. Some factors that affect readability are:

- the choice of words in the text (lexical level)
- the text surface properties (syntactic level)
- the text content, the text perspective, the voice, and the coherence (semantic level)

The syntactic structure makes it easier for the reader to memorize a text and the semantic structure to understand the text (Platzack, 1989). When we read, processing occurs on several levels concurrently; semantic, syntactic and textual information are extracted in parallel. In the following subsections, we describe properties of a text that influence comprehension during reading on lexical, syntactic and semantic level.

2.3.1 Lexical level

Anything that has to do with the vocabulary in the text belongs to the lexical level, according to Melin and Lange (2000). Compounds and long words exert our short term memory and make reading and understanding more difficult (Melin, 2004). Abstract words, unfamiliar words, jargon and technical language demand a clear context to become understandable. Homonyms, words with double meaning, can cause interpretation problems, but the context helps the reader to decide what is intended (Platzack, 1974).

To measure readability based on the factors on lexical level, it is possible to design a readability formula or use an already existing one. In a readability formula, language variables that can be used are long words, abstract words, unfamiliar words, and so forth. These language variables can be counted and compared to sentence factors. That in turn describes the complexity of a sentence, in order to determine a texts reading ease (Platzack, 1974).
2.3.2 Syntactic level

A sentence that has a complicated syntax takes longer to read than a sentence with simpler syntax, if they are otherwise equivalent (Platzack, 1989). An unclear syntactic structure makes reading more difficult because it is more difficult to interpret the meaning from the text. Here we give some factors that make reading more difficult.

According to Platzack (1974), a standard readability question is: How long should a sentence be to be maximally readable? His answer is that neither too long nor too short sentences are good. According to the sentence length hypothesis presented by Platzack (1974), a text with an average sentence length of approximately 13 words is easier to read than a text with an average sentence length of less than 9 words.

Long sentences become hard to read when they have a complicated sentence structure (Gunnarsson, 1989). It is the combination of words in the sentence that makes it harder to read, not the fact that the sentence consists of many words (Platzack, 1989). Longer sentences, however, often have more complicated structure and less redundancy (Gunnarsson, 1982).

According to Gunnarsson (1989), punctuation marks and form words; e.g., prepositions and conjunctions, give the reader clues about the syntactic structure, and such clues makes it possible to read faster.

Another type of sentence structure that makes reading more difficult is nesting of subordinate clauses in sentences, according to Gunnarsson (1989). An example of this, according to Gunnarsson (1989), is:

Example 2.1 *John, whom June, whom Paul prefers, detests, loves Mary.*

Gunnarsson (1989) also states that passive sentences are more difficult to read than active sentences. Platzack (1974) does not completely agree with this statement. According to him, sentences in passive form are not always more difficult to read than sentences in active form. In 1966, Dan Slobin found that passive form is only more difficult when the semantic relation between the subject and the object is unclear; i.e., when it is hard to directly realize who/what is the subject and who/what is the object (as cited in Platzack, 1974). The following examples illustrate this:

Example 2.2 *The dog ate the cake.*

Example 2.3 *The cake was eaten by the dog*

Example 2.4 *The horse bit the cow.*

Example 2.5 *The cow was bitten by the horse.*
In 2.2 and 2.3 it is easy to understand who the subject is and the sentences are equally simple to understand (Platzack, 1974). In 2.4 and 2.5 the action could be performed by the noun that is the subject and also the one that is the object. That is why the passive form is more difficult to understand than the active (Platzack, 1974).

2.3.3 Semantic level

A few types of semantic relations associated with a text are: cohesion, redundancy and thematic roles. Halliday and Hasan (1976) state that cohesion is in some ways the most important concept, since it is common to all kinds of text and what makes a text a text. Redundancy is relevant to study because increased redundancy generally improves readability, according to Horning (1991). Thematic roles play a major role during disambiguation during reading.

Cohesion and redundancy

Readers rely on two factors in a text to get the meaning: cohesion and psycholinguistic redundancy (Horning, 1991). The first factor, cohesion, has been shown to play a central role in reading, according to Horning (1991). Halliday and Hasan (1976) define cohesion in the following way:

The concept of cohesion is a semantic one; it refers to relations of meaning that exist within the text, and that define it as a text.

(Halliday and Hasan, 1976, p.4)

Cohesion connects a string of sentences to form a text rather than a series of unrelated statements (Halliday and Hasan, 1976). The unit of analysis for cohesion is the cohesive tie. Cohesive ties can be categorized and counted (Halliday and Hasan, 1976).

Ties can be anaphoric or cataphoric; i.e., refer backward or forward, and located at both the sentential level and the unit of language larger than a sentence (Halliday and Hasan, 1976). According to Halliday and Hasan (1976), cohesion is classified under two types: grammatical and lexical. Grammatical cohesion is expressed through the grammatical relations in text such as ellipsis and conjunction. Lexical cohesion is expressed through the vocabulary used in text and the semantic relations between those words. Halliday and Hasan (1976) proposed a way to systematize the concept of cohesion by classifying it into five major categories of cohesive ties that occur in text: reference, substitution, ellipsis, conjunction, and lexical cohesion. Halliday and Hasan (1976) state that:
Each of these categories is represented in the text by particular features—repetitions, omissions, occurrences of certain words and constructions—which have in common the property of signalling that the interpretation of the passage in question depends on something else. If that 'something else' is verbally explicit, then there is cohesion. (Halliday and Hasan, 1976, p.13)

The Table 2.1 shows example of each of the five different cohesive ties given in Halliday and Hasan (1976).

<table>
<thead>
<tr>
<th>Type of cohesion tie</th>
<th>Example</th>
</tr>
</thead>
</table>
| Reference            | John has moved to a new house.  
                        |          |
|                      | He had it built last year.      |
| Substitution         | Is this mango ripe?            |
|                      | It seems so.                   |
| Ellipsis             | Has the plane landed?          |
|                      | Yes it has <...>.              |
| Conjunction          | Was she in a shop?             |
|                      | And was that on the counter really a sheep? |
| Lexical cohesion     | Why does this little boy wiggle all the time? |
|                      | Girls don’t wiggle.            |

**Reference** is divided in three categories in English, according to (Halliday and Hasan, 1976): personal, demonstrative, and comparative reference. The example of reference given in Table 2.1 is a personal reference. Personal references are often nouns, pronouns, determiners that refer to the speaker, the addressee, other persons or objects, or an object or unit of text (Halliday and Hasan, 1976). Demonstrative reference are determiners or adverbs that refer to locative or temporal proximity or distance, or that are neutral (Halliday and Hasan, 1976). Comparative reference are adjectives or verbs expressing a general comparison based on identity, or difference, or express a particular comparison (Halliday and Hasan, 1976).

**Substitution** is replacement of one item in the text by another (Halliday and Hasan, 1976). In English, there are three types of substitution: nominal (items that occur as substitutes: one, ones, same), verbal (do), and clausal (so, not). The example of substitution given in Table 2.1 is a clausal reference.

**Ellipsis** is omission of an item and can be interpreted as a form of substitution in which the item is replaced by nothing (Halliday and Hasan, 1976).
Conjunction is divided in four categories: additive, adversative, causal, end temporal (Halliday and Hasan, 1976). By using the word and, an additive cohesive tie, in the example sentence in Table 2.1, the author has helped the reader to link one sentence to another.

Lexical cohesion is the cohesive effect achieved by the selection of vocabulary (Halliday and Hasan, 1976). There are three classes of lexical ties, according to Halliday and Hasan (1976): general noun, reiteration, and collocation. Reiteration involves the repetition of an lexical item, at one end of a a scale. The repetition of the same lexical item strengthens the text cohesion. Collocation is achieved through the association of lexical items that regularly co-occur and an example of his is given in Table 2.1 (Halliday and Hasan, 1976).

If the levels of cohesion is increased in text improves reading comprehension as measured by reading time and recall of content (Horning, 1991).

According to Horning (1991), the best definition of redundancy in reading comes from the work of Frank Smith. Smith, as cited in Horning (1991), says that:

[Redundancy consists of] information that is available from more than one source. In reading, redundancy may be present in the visual information of print, in the orthography, the syntax, the meaning or in combinations of these sources. Redundancy must always reflect non-visual information; prior knowledge on the part of the reader permits redundancy to be used.² (Horning, 1991)

Cohesion and redundancy are related to each other in more than one way (Horning, 1991). Cohesive ties create redundancy (Horning, 1991).

Thematic roles

A valuable classification of words for disambiguating purposes is thematic roles³. Just as each word, by itself, belongs to a part of speech⁴, and, depending on its syntactic, grammatical function, is a clause element⁵, it also has a semantic function. This semantic function is called thematic role. A thematic role is the underlying relationship that a participant has with the main verb in a clause (Payne, 1997b).

²Note that the extra characters in the citation exist in the original online journal.
³also known as semantic roles
⁴also known as word class
⁵also known as a component part of a sentence
Grammatical relations; e.g., subject and object, are morphosyntactic, whereas thematic roles; e.g., agent, patient, and instrument, are conceptual notions (Payne, 1997a). Semantic roles do not correspond directly to grammatical relations. Table 2.2 illustrates an example of the varying thematic roles that a subject can have (Payne, 1997a).

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Grammatical relation</th>
<th>Semantic role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob opened the door with a key.</td>
<td>Bob = SUBJECT</td>
<td>Bob = AGENT</td>
</tr>
<tr>
<td>The key opened the door.</td>
<td>The key = SUBJECT</td>
<td>The key = INSTRUMENT</td>
</tr>
<tr>
<td>The door opened.</td>
<td>The door = SUBJECT</td>
<td>The door = PATIENT</td>
</tr>
</tbody>
</table>

**Information density**

Complex sentences are tiresome for most readers in that they reduce reading speed and comprehensibility (Melin, 2004). Complexity is often connected to information density. To express oneself concisely is a good thing but only up to a certain point.

If the text does not contain any information at all, i.e., it is just nonsense, the reader will not understand anything. And if we have maximal information packing the comprehension is also zero. This depends on the fact that reading is an encounter between the new information that we read and our background knowledge where we try to insert the new knowledge (Melin, 2004). Information density is measured by the nominal quotient (NQ), which is a quotient between nominal parts-of-speech (information carriers): nouns, prepositions and participles, and verbal parts-of-speech: verbs, pronouns and adverbs (Melin, 2004). This is the formula for calculating the NQ:

\[
NQ = \frac{nouns + prepositions + participles}{pronouns + adverbs + verbs}.
\]

The normal score for NQ is 1.0; i.e., the score for newspapers and textbook for high school (Melin and Lange, 2000). Easy readable texts should not have higher NQ than 1.0.
2.4 Comprehensibility measures

The comprehensibility measure seeks to address the question: “Is the text understandable?”. The most common ways to assess text comprehension, according to Garnham (2003), are memory measures and on-line measures.

Memory measures include the following methods:

- recall of text, which invite a subject to reproduce—orally or in writing—the content and structure of the whole or part of a text

- reading speed procedures, which assume that a subject can read easy texts faster than difficult ones

- rating procedure, which require a subject to assess his own comprehension of a text on a scale of low to high

- forced-choice procedure, which is a subjective judgment of comprehensibility on a sentence-by-sentence basis. The reader is asked to assign a 1 or a 0, depending on whether a sentence was comprehensible or not. The final comprehensibility score for the text is then the total number of comprehensible sentences divided by the total number of sentences (Miller et al., 2001).

- question procedures, which require subjects to answer questions about a text’s content in order to investigate their comprehension of the text

- action procedures, which require a subject to read a text with instructions and then carry out the prescribed actions

- thinking-aloud procedures, which require a subject to verbalize the process of decoding the meaning of a text

- counting the number of phone calls technical writers receive from translation companies where they need clarification of a text (Haller and Schütz, 2001).

The other main type of methods for comprehensibility measures is on-line measures, which includes reading time from screen or eye-movement procedures. Generally speaking, on-line methods consist of analyzing the cognitive activity in progress. This is done instead of, or in addition to, the activity outcomes; e.g., the individuals performance on a post test.
2.4.1 Comprehensibility on small versus large texts

Halliday and Hasan (1976, p.1) say that “The word text is used in linguistics to refer to any passage, spoken or written, of whatever length, that does form a unified whole.” When a reader reads a passage of language which is longer than one sentence, he can easily decide if the passage forms a unified whole or is just a collection of unrelated sentences (Halliday and Hasan, 1976). It is important to realize the difference between the two in order to understand how you can measure small and large texts.

Methods for measuring comprehensibility and readability vary depending on the size of a text. If the text is small, it is easier to “read into” the meaning, or to infer it from the context. If the text is large, you will have to measure how simple it is to understand the text as a whole; i.e., the extent to which valid information and inferences can be drawn from different parts of the same document.

Comprehension of a large coherent text is something more than just the sum of understanding of its individual sentences, according to Gunnarsson (1982). She continues by explaining that text understanding is a constructive process where the reader builds the descriptions of the entirety based on the text and also from his structure of reality.

2.5 Readability measures

Large parts of readability research has not been affiliated with psychological nor linguistic theories, according to Gunnarsson (1982). Instead, it has been controlled by the practical purpose to develop an instrument for deciding how difficult a text is to read (Gunnarsson, 1982). The most commonly used instruments for measuring reading ease are different kinds of readability indexes (Gunnarsson, 1982). The technical writer William H. DuBay(2004) wrote the following lines about readability formulas used today:

> The principles of readability are in every style manual. Readability formulas are in every word processor. What is missing is the research and theory on which they stand. (DuBay, 2004, p.1)

Researchers have used at least two approaches to assess readability of documents (Shelby, 1992). The first is based on the notion that “readability is in the mind of the reader”, and the second approach measures readability through document analysis, which is usually numerical (Shelby, 1992).
2.5.1 The assessment test

One method for assessing readability is to let a group of people, experts or others, assess texts relative degree of difficulty (Björnsson, 1968). The judgement can be done by comparison in pair; i.e., the texts are compared two and two. One text version is compared with a prototype. According to Björnsson (1968), this measure is the best for testing readability because it means that people are asked about how they experience texts and measuring readability is about the experienced difficulty degree.

2.5.2 The cloze procedure

In 1953, Wilson L. Taylor of the University of Illinois introduced the CLOZE PROCEDURE as a measure of readability (DuBay, 2004). In order to perform the cloze procedure a cloze test is constructed which uses a text with selected words deleted and replaced with underlines of the same length (DuBay, 2004). The percentage of words correctly entered is the cloze score. The lower the score, the more difficult the text. Cloze tests can be used either at sentence-level or cross-sentence level. The cloze procedure measures a reader's comprehension of a text.

2.5.3 The readability indexes

In the 1920s, educators discovered a way to use vocabulary difficulty and sentence length to predict the level of difficulty for a text (DuBay, 2004). They embedded this method in readability formulas, which have been used for 80 years now (DuBay, 2004).

Research on the readability formulas was something of a secret until the 1950s, when writers like Rudolf Flesch and George Klare popularized the reading formulas by presenting the research that supported them (DuBay, 2004). By the late 1980s, there were 200 formulas and a thousand studies published on readability formulas proving their theoretical and statistical validity (DuBay, 2004).

READABILITY INDEXES measure text features that promote readability and can be mathematically measured (DuBay, 2004). They can not measure how comprehensible a text is; i.e., they do not help us evaluate how well the reader will understand the ideas in the text. However, readability formulas are fast and economical to implement (DuBay, 2004). Readability indexes can only be used for larger texts since they use mathematical formulas based on the number of words, and are not appropriate for measuring readability on the sentence level. Readability indexes help improve the text on the
level of words and sentences, that are the first causes of reading difficulty (DuBay, 2004). Some commonly used readability indexes are the Flesch-Kinkaid Index, Rudolf Flesch’s Reading Ease Formula, SMOG readability formula and the Fog Index. One of the most common indexes, Flesch’s reading ease formula, is described below.

The Flesch’s Reading Ease Formula

One of the most widely used and well-known formulas for assessing readability of English texts is the Flesch’s Reading Ease Formula (Hargis, 2000). Reading ease (RE), according to this formula, is defined as

\[ RE = 206.835 - 0.846WL - 1.015SL \]

where WL (word length) is the number of syllables per 100 words, and SL (sentence length) is the average number of words per sentence (Hargis, 2000). The RE ranges from 0 to 100 and is based upon average sentence length and average syllable density per 100 words. High index scores represent high readability predictions. The standard score is 60–70 and the users of the formula should try to reach that readability score (Lemos, 1985). Table 2.3 describes the reading ease categories according to the Flesch reading ease index.

<table>
<thead>
<tr>
<th>Index score</th>
<th>Readability</th>
</tr>
</thead>
<tbody>
<tr>
<td>0–30</td>
<td>Very difficult</td>
</tr>
<tr>
<td>30–50</td>
<td>Difficult</td>
</tr>
<tr>
<td>50–60</td>
<td>Fairly difficult</td>
</tr>
<tr>
<td>60–70</td>
<td>Standard</td>
</tr>
<tr>
<td>70–80</td>
<td>Fairly easy</td>
</tr>
<tr>
<td>80–90</td>
<td>Easy</td>
</tr>
<tr>
<td>90–100</td>
<td>Very easy</td>
</tr>
</tbody>
</table>

Applying the Flesch’s Reading Ease Formula to other languages does not deliver good results because of the different language structure. For example, Spanish texts have much higher syllable counts than English texts.

Flesch-Kincaid Grade Level Formula

The most common readability formula is Flesch-Kincaid Grade Level formula which converts the Reading Ease score to a U.S. grade-school level. Reading ease (RE), according to this formula, is defined as

\[ RE = 0.39ASL + 11.8ASW * 15.59 \]

where ASL (average sentence length) is the number of
words divided by the number of sentences, and ASW (average number of syllables per word) is the number of syllables divided by the number of words. Standard writing approximately equates to the seventh-grade to eighth-grade level.
Chapter 3

Translatability

A good translation is dependent on many factors, such as the skill of the translator and the similarity between the languages. In this thesis, the focus of translatability is on how the writing style of the source text affects the quality of the translation.

This chapter is written as a background to translation, with the intent of giving an overview of the problems of translation and how to define and measure translatability.

3.1 The goal of translation

What is a good translation of a piece of text? When translating a poem, it is necessary that the beauty is preserved, and when translating an instruction, there must be no added ambiguities. In both cases, the intention of the original writer should be transferred from one language to another; the poem should call to the same feelings, and the instructions should tell the users to perform the same operations.

Newmark (1988a) writes that translation is a craft, consisting in the attempt to replace a written message or statement in one language with the same message or statement in another language.

The translation debate during the nineteenth century focused on whether a translation should incline towards the source or the target language (Newmark, 1988a). If it is close to the source languages, it is likely to be faithful, literal, and form centered. The opposite—closeness to the target language—yields a translation that is beautiful, free, and content centered (Newmark, 1988a).

Nowadays, there is a wide agreement that the main aim of the translator is to, as nearly as possible, produce the same effect on his readers as produced
Kay (2004) expresses the idea of translation as twofold: the source and result shall tell the same story and they shall have corresponding parts. This means that the reader should end up in the same mental state and should pass through the same sequence of mental states. Similar readers should acquire the same knowledge and learn, consider, and question the same things, in the same order (Kay, 2004).

Newmark (1991) discusses the translator’s required creativity. When translating a serious text, the translator must find a compromise between strict accuracy and conciseness. When translating low quality informative texts, the translator must be creative in another sense—by turning bad into good writing (Newmark, 1991).

The most successful translation, according to (Newmark, 1991), is the one that can convincingly transfer the most important components of the source text into the target text. Translating poetry is the most extreme case, where many aspects such as meter, rhythm, and sounds are important (Newmark, 1991).

Newmark (1991) summarizes the discussion of the goal of translation as a being both ends and means. The translation must be both accurate and creative.

I would say that both ends and means are always important, that the end never justifies inappropriate means (the writing), and that for a serious text its end and means often prescribe those of the translation (it being of greater value) which may require an unaccustomed humility from the translator. (Newmark, 1991, p.10)

### 3.2 The translator’s comprehension

In order for the translator to be able to transfer the author’s intention when translating a piece of text, the translator must be able to grasp that intention. The following words by Nida (2001) illustrates exactly this:

Clarity in understanding the source text is the key to successful translating into a receptor language. Translators do not translate languages but text. (Nida, 2001, p.3)

Since the translator might not be an expert in the topic of the text and probably does not belong to the target group, it is extra important that the
author considers the comprehensibility of his text before submitting it for translation; see Chapter 2 for a general discussion of this.

What differs from general comprehensibility is that the translator does not have to understand each technical term in order to produce a good translation, since he could look these up in a dictionary. However, if the term has more than one possible translation, depending on its intention, the translator will have to try to figure out the correct interpretation, which can be more or less difficult. This is further discussed in Section 3.5.

3.3 Target language independence

For obvious reasons, it is easier to translate between related languages than between languages with relatively little in common. The translation problem depends on the languages involved, so the translatability of a text is different depending on the target language. However, it would be interesting to be able to talk about translatability in a way that is target language independent.

Are there any constructions that are generally difficult for a translator to handle? If so, it makes sense to put different requirements on different texts depending on whether they are intended to be translated, no matter into which languages.

Allen (1999b) writes about the distinction between texts destined for translation and texts that someone later on have chosen for translation. He means that it is indeed possible to implement writing principles that will improve the general translatability of the texts. We will come back to this kind of principles in Chapter 5.

In Section 3.5, a number of common translation problems are discussed with target language independence kept in mind.

3.4 Semantic or communicative translation

One of the main contributions to translation theory by Newmark (1991) is the classification of communicative and semantic translations. In short, a communicative translation is reader-centered, effective, and focused on the message, while a semantic translation is author-centered, informative, and focused on the meaning (Newmark, 1991).

A vast majority of texts should, according to Newmark (1991), be translated communicatively. The exceptions are works where the language of the writer is as important as the concept. In these cases, the freedom of the translator is limited—the translation must be faithful and the translator has
no right to improve or correct the text. This does not make semantic translation easier; in fact Newmark (1991) describes semantic translation as an art, whereas communicative translation is a craft.

Newmark (1991) stresses that in both communicative and semantic translation, the best translation is always the word-for-word translation, provided that the equivalent effect is secured compared to the original. That means that there are no excuses for unnecessary synonyms or elegant variations.

It is interesting to compare these two approaches to translation with Bühler’s text functions. See Section 4.5.7 for a discussion of this.

3.5 Common translation problems

A translator is faced with many problems when translating a text. In this section we discuss the problems of understanding three aspects of the text: the intended meaning of the words, the relations between the words, and the actual message of the text. Finally, we discuss references made across the text, which, as we will see later, is especially troublesome for machines to handle.

3.5.1 Lexical ambiguity

When a word can have more than one meaning, we have what is called a lexical ambiguity. The translator must then try to figure out which meaning is intended.

It is common in English that one word can belong to several parts of speech. This is most obvious with nouns, which often can be used as verbs without changing the way that they are written.

Example 3.1 Elevators use electricity, which makes elevator use dangerous during fires.

To be able to choose the correct meaning of the two occurrences of “use” in Example 3.1, it suffices to examine the phrase in which the words occur; only one interpretation will make the sentence grammatical. This kind of examination is called SYNTACTIC ANALYSIS (Arnold et al., 1994).

Example 3.2 shows how the sentence can be rewritten with the ambiguity removed.

Example 3.2 Elevators utilize electricity, which makes elevator usage dangerous during fires.
Sometimes, a syntactic analysis alone is not enough to choose the correct meaning. However, with some knowledge of the topic and the context, probably only one of the syntactically correct meanings would make sense. This is called SEMANTIC ANALYSIS and requires that the translator is able to examine and understand the context of the word (Arnold et al., 1994).

### 3.5.2 Syntactic ambiguity

When a phrase or a sentence can have more than one structure, it has SYNTACTIC AMBIGUITY\(^1\) (Arnold et al., 1994). With structure, we mean grammatical representation with a hierarchy of words, phrases and clauses. Here are some examples of different kinds of syntactic ambiguity (Kim, 2002):

**Example 3.3** The little boy hit the child with the toy.

**Example 3.4** We need more intelligent leaders.

**Example 3.5** The mother of the boy and the girl will arrive soon.

Example 3.3 belongs to one of the most commonly discussed categories, namely PREPOSITIONAL PHRASE ATTACHMENT. The problem lies in that the prepositional clause “with the toy” could describe the child being hit, called NOUN ATTACHMENT, or describe the hitting, called VERB ATTACHMENT (Hindle and Rooth, 1991).

Examples 3.4 and 3.5 are other examples of syntactic ambiguity. In Example 3.4, the question is whether “more” is an adjective or an adverb; i.e., if the number of leaders or their intelligence should be increased, and in Example 3.5, the span of the conjunction is unknown; is the mother also a mother of the girl?

Naturally, ambiguity makes texts harder to comprehend, but it affects translatability even further. The meaning can be perfectly clear for the author and the intended reader; however, the translator, who might not be an expert in the domain, could choose the wrong interpretation causing the translation to confuse or even deceive the reader (Arnold et al., 1994).

**Compounds**

Compounds is another case of syntactic ambiguity in English. A compound is a combination of two or more words which functions as a single word (Arnold et al., 1994).

\(^1\)also known as STRUCTURAL AMBIGUITY
How compounds should be written in English is a debated issue. There are three main ways of constructing compounds. Darling (2001) gives the following examples:

**closed form** childlike, keyboard, notebook

**hyphenated form** over-the-counter, six-pack, mass-produced

**open form** post office, real estate, full moon

When a new compound is formed out of two nouns, the open form is usually used. Over time, when the compound gets more commonly used, it has a tendency to evolve into the closed form, sometimes through the hyphenated form (Darling, 2001).

One problem with compounds is that the relation between the words are not always clear. Arnold et al. (1994) give the following examples:

**Example 3.6** student film society

**Example 3.7** satellite observation

Example 3.6 is a problem of grouping; it can mean a society for student films just as well as a film society for students. Example 3.7 is a problem of how the words are related; it can mean both observation by satellite and observation of satellites.

Apart from the general problem of translating ambiguous text, there is a problem in that not all languages allow the creativity of compounds, which requires reformulations of these constructions (Arnold et al., 1994). Compounds of many words can thus require a considerable effort when being translated.

### 3.5.3 Contextual ambiguity

On a higher level than just the basic meaning of the words and sentences is the context. Nida (2001) writes in an example that when a person buys a large home in an exclusive neighborhood, it can serve as a place to house a large family, a way of showing off his wealth, a place for entertaining large numbers of guests, or a good investment. The reason for an event is part of the context, just as who does what, when, and where (Nida, 2001).

It is important for the translator to get a good view of the context before translating a text. This helps in choosing translated sentences that do not only comply in their literal meanings, but also in their nuances and associations (Nida, 2001). Literal meaning is known as **denotation**, and what the word or sentence suggests beyond that is referred to as **connotation** (Britannica Student Encyclopedia, 2004).
3.5.4 Anaphora

The anaphora phenomenon can be considered one of the most difficult problems in natural language processing (Peral and Ferrández, 2003). It is especially troublesome for machines, since it requires semantic knowledge, but even human translator can have difficulties with anaphora. Graeme Hirst gives the following thorough definition of anaphora:

[Anaphora is] a device for making an abbreviated reference (containing fewer bits of disambiguating information, rather than being lexically or phonetically shorter) to some entity (or entities) in the expectation that the receiver of the discourse will be able to disabbreviate the reference and, thereby, determine the identity of the entity. (Hirst, as cited in Peral and Ferrández, 2003, p.117)

The reference is called anaphor and the referenced entity is its antecedent. Peral and Ferrández (2003) give an enlightening example; see Example 3.8.

Example 3.8 Mary went to the cinema on Thursday. She didn’t like the film.

In Example 3.8, “she” is the anaphor and “Mary” is the antecedent, since “she” is referring to “Mary”.

There are various types of anaphora, but the most common variant is where a pronoun references a noun phrase, as in Example 3.8. This type is called pronominal anaphora (Peral and Ferrández, 2003).

In order to understand a text, it is necessary that the reader can figure out the antecedents of the anaphors; this process is called anaphora resolution (Peral and Ferrández, 2003).

Translating anaphora can be easy or very difficult depending on the languages involved. With closely related languages, it can be enough to translate the anaphor without resolving its antecedent. In many cases, it is not that simple. When translating into languages with multiple genders and where the gender affects the pronouns, it is vital that the translator can resolve the pronominal anaphora (Peral and Ferrández, 2003).

Anaphora often makes texts ambiguous, and it is then necessary to use syntactic and semantic knowledge to resolve it. Syntactic knowledge tells us that “she” must refer to a feminin noun, but we need semantic knowledge to resolve examples 3.9–3.11, taken from Peral and Ferrández (2003).

Example 3.9 The monkey ate the banana because it was hungry.
Example 3.10 The monkey ate the banana because it was ripe.

Example 3.11 The monkey ate the banana because it was tea-time.

In Example 3.9–3.11, the anaphor “it” refers to “monkey”, “banana” and the abstract notion of time, respectively (Peral and Ferrández, 2003). This makes the translation of “it” different in these cases when translating to, for instance, Spanish. In order to translate this correctly, knowledge of monkeys, bananas, and time is required.

As pointed out before in this chapter, when semantic knowledge is required, it is possible that the translator, who may not be an expert in the domain, lacks that knowledge and makes an incorrect translation.

However, there is another translation problem to be aware of. The translation can, even if it is correct, in fact introduce ambiguity.

Example 3.12 The boy and the girl are in the same pool. He is to the left of her.

Example 3.13 The mice and keyboards are on the same shelf. They are to the left of them.

Example 3.13 is a possible result of a translation, where “they” and “them” have become ambiguous. The reason for this could be that the text is translated from a language such as French, where mice and keyboards have different genders and thus have different pronouns. The original was unambiguous, just as Example 3.12 is. This syntactic distinction is thus lost in the translation and ambiguity is introduced.

3.6 Machine translation

Machine translation is, not very surprisingly, translation performed by machines. Since the early days of computers, using machines for translations has been an exciting idea, and it still is.

There are various reasons to advocate translation in general. Commercially, it has been shown that instruction manuals in a native language makes products more attractive (Arnold et al., 1994). However, translation is also important for social and political reasons in geographical regions where multiple languages are spoken (Arnold et al., 1994), not to mention translation of safety instructions. In many of these cases, the volumes are large and it is important to get a translation quickly. This is often impossible for the relatively few human translators to handle unless they get help from automation (Arnold et al., 1994).
3.6.1 Adapting to machine translation

It is important to consider that most translation agencies use machine translation to some extent. The typical translation process is, according to Arnold et al. (1994):

- Preparation
- Machine translation
- Revision

In the revision step, a professional translator does the post-editing of the texts, where he corrects the mistakes that the machine translation has made. However, it is important to realize that this can take a lot of time compared to the machine translation step. Therefore, if the source texts can be written in a way that is easily handled by machine translation, the time and cost for translation will decrease significantly (Arnold et al., 1994).

3.6.2 Typical machine translation problems

Many problems of translation are shared between human translators and machine translation programs. Languages with little in common are hard to translate between, because there might not exist a word with the same intension in the target language (Arnold et al., 1994). In these cases, humans have an advantage in their creativity; they can rewrite the text so it is easier to translate. However, this is essentially a preparation for translation and could be done regardless of whether humans or machines perform the actual translation.

The most troublesome issue for machine translation is, without doubt, semantics. In order to be able to translate anything, the machine translation program must in a sense know everything. Arnold et al. (1994) illustrates this by suggesting translation of Example 3.14 into French:

Example 3.14 negatively charged electrons and protons (Arnold et al., 1994, p.5)

The ambiguity in this sentence is a case of conjunction span, as discussed in Section 3.5.2, and knowledge that protons are not negatively charged is required for a correct translation.

Arnold et al. (1994) writes that in order for machine translation systems to perform well, the texts must have few ambiguities or the text must belong
to a domain, where some knowledge is already programmed into the system. Naturally, a combination of these requirements are preferred.

The anaphora problem described in Section 3.5.4 is a similar problem and very difficult for computers to handle (Peral and Ferrández, 2003).

Another difficulty for machines is headings, which are often phrases, but treated by the computers as clauses (Brekke and Skarsten, 1987).

**Example 3.15** Heading 1: Check boxes

Example 3.15 should probably be interpreted as a noun compound, signalling that the following chapter will deal with check boxes. However, if it had been part of a list, “Check boxes” could be interpreted as an instruction. This shows that information about text elements such as headings and lists could be useful to a machine translation system. Marking text segments as instructive or descriptive would further simplify these choices (Brekke and Skarsten, 1987).

### 3.7 Translation memories

One of the most successful inventions in the area of computer aided translation is, without doubt, the translation memory technology. Over the past 20 years, it has been a major focus for improving translation (Allen, 1999a).

Translation memories are typically integrated into the translation environment and can be used together with machine translation or human translation. The idea is that a copy of each translated sentence is stored in the memory. Then, if the system encounters an identical sentence, it can automatically retrieve and suggest the translation from the memory (Arnold et al., 1994).

Translation memories can save a lot of time and money for a company that translate texts within a narrow domain. Translation memories have been reported to handle as much as 90% of the sentences for organizations which re-translates revised manual editions (Arnold et al., 1994). However, an important question is why these revised texts were sent for translation as a whole. If the documentation system would have been able to keep track of changes, only the new sentences would be sent for translation, which even further would reduce time and cost.

Nonetheless, translation memories are efficient if the language is limited and standardized, which increases the chances of writing a sentence that has been written and translated before (Arnold et al., 1994).

It should be noted that there are some pitfalls related to translation memories as well. If a sentence is wrongly translated, there is an obvious
risk that the same error will be made again. Translation memories also
discourage introducing new terms or ways to write text. Even if this often is
a good thing, improvement to writing guidelines can be expensive, since the
translation memories will not handle the new constructions. The result of
using translation memories could thus be that the documents become more
consistent, even if they become consistently hard to read.

3.8 Parameters of correspondence

There are many ways to classify how true a translated text is its original.
Newmark (1988a) has defined a number of parameters of the source text that
the translator has to consider and try to copy. We here list these parameters,
and in parenthesis we give possible extreme values as presented by Newmark
(1988a):

- formality (frozen – uninhibited)
- feeling or affectivity (overheated – deadpan)
- generality or abstraction (popular –opaquely technical)
- evaluation:
  - morality (good – bad)
  - pleasure (nice – nasty)
  - intensity (strong – weak)
  - dimension (wide – narrow)

It is impossible to find a translated text that exactly matches the original with
regard to the above factors everywhere, but Newmark (1988a) has proposed
the idea that collocations, idioms, metaphors, proverbs and so forth should
be equally frequent in the target language as in the source language. This
does not seem to imply that an idiom must be translated to an idiom, but
merely that the resulting text should be on the same level as the source text
with respect to the parameters above.

Finally, Weightman (as cited in Newmark, 1988a) has made an interesting
point about preserving style:

A good writer’s use of language is often remote from, if not at
cross purposes with, some of the conventional canons of good
writing, and it is the writer not the canons that the translator
must respect. (Weightman, as cited in Newmark, 1988a, p.8)
3.9 Translatability measures

This chapter has not yet contained any numbers or formulas for calculating translatability. Spyridakis et al. (1997) mention that there is very little written about measuring translatability, and we have not been able to find any general methods for doing this.

Evaluating the actual translation performed on a specific document seems to be an easier task, but Ingo (1991) writes that it does not exist an objective model as basis for evaluation of translations, even though it should be possible to point out some central aspects, which should be regarded in such evaluations. But even if there are no commonly used methods, Ingo (1991) continues, several suggestions have been presented in scientific literature for translators.

The resulting translations can be evaluated with regard to resulting correctness, comprehensibility, and preservation of meaning and style. (See further Chapter 2 and Chapter 4 on comprehensibility and style.)

From an industrial point of view, the cost for producing a good enough translation of a document is a highly relevant measure of translatability. The cost is, naturally, dependent on how well the source text is written, but other factors play important roles as well, such as similarity to already translated material and adjustment to translator tools (Arnold et al., 1994).

We will here shortly describe two studies: one made by Spyridakis et al. (1997), in which a method is invented for measuring general translatability, and one by Bohan et al. (2000), in which translation made by a machine translation system is evaluated. We will also dedicate a section to a quantitative measure widely used for evaluating machine translation quality today—the Bleu method.

3.9.1 Study 1: Measuring general translatability

In their study, Spyridakis et al. (1997) constructed a method for measuring and comparing general translatability. They wanted to investigate the potential of a text to be translated easily and with a high quality result.

In order to measure the translatability of a document, they chose three different languages—Chinese, Spanish, and Japanese—and picked university students that were native speakers of these languages respectively and with good English skills. One reason for using this group of people, apart from not costing much money, is that Spyridakis et al. (1997) believed that university students would represent novice translators well.

The translator subjects were instructed to preserve the meaning and style of the original document, mark unknown words and leave English words
that they believed should not be translated. The subjects also provided
demographic information, which would be used to ensure that the results
were not dependent on such factors.

The original document was also translated by professional translators,
who were familiar with the document type. These translations served as
baseline translations and were designed to exemplify good translations.

An importance rating was also made by a professional editor within the
domain, where all terms, sentences and sections were marked as “more im-
portant” or “less important”.

Three native speakers from each language, selected for their good com-
unication skills, were then assigned as raters of the translations made. The
raters then judged the translations on the following measures:

- Accuracy of Translation
- Style Match with the Original Document
- Ease of Comprehension
- Number of Mistranslations
  - Minor\(^2\)
  - Major\(^3\)
- Number of Omissions
  - Minor
  - Major

The greatest dilemma in the study was, according to Spyridakis et al. (1997),
how to measure ease of translation. They used the measure of style match to
indirectly assess the ease of translation, since if a document is easy to trans-
late, the style of the original document is more likely to be preserved. They
also reasoned that a more comprehensible document might result when the
translation task is easy. This further stresses the close relationship between
comprehensibility and translatibility.

In addition, Spyridakis et al. (1997) asked the subjects how easy the task
was, but did not get many usable responses.

The scores on the different translations of the same document were then
analyzed to get statistical data such as average values and variance. These

\(^2\) Mistranslation of parts marked as “less important”.

\(^3\) Mistranslation of parts marked as “more important”.
values were then used to compare two documents, to investigate if one document was significantly more translatable than the other.

Another goal of Spyridakis et al. (1997) than simply measuring the translatability of the involved documents, was to investigate how a method for measuring translatability could be designed. They were satisfied with the design, and we believe that it is interesting for future research about translatability.

3.9.2 Study 2: Evaluating machine translatability

Bohan et al. (2000) have performed a study of translation quality of machine translation systems. Their purpose was to identify the most costly grammatical and lexical weaknesses in order to find out where to improve their systems. Assuming that their findings are applicable for other machine translation systems, we can look at their results as a list of grammatical and lexical constructions that are troublesome for machine translation in general.

The method chosen by Bohan et al. (2000) was corpus-based. They selected texts from the Internet and from a corpus cd in order to get a true distribution of linguistic phenomena as opposed to what would have been found in a prepared test suite. A BLACK BOX VIEW was used, which means that they did not study how the text was processed inside the system, but only the resulting translation. According to Galliers and Sparck Jones (1993), this permits distinction of objective, quantitative measures of numbers of errors in translation, based on subjective measures of quality; i.e., understandability ratings by humans.

The corpus texts passed through the following suite, where only the texts that got the worst score continued to the next step.

1. Evaluate translation on text level based on
   - understandability
   - grammaticality
   - lexical correctness

2. Evaluate translation on sentence level based on
   - preservation of meaning
   - grammatical correctness

3. Compare with translation from another system
4. Identify the problematic linguistic phenomena

Bohan et al. (2000) used the third step to select only those texts where their system performed significantly worse than the other system. They thus filtered out sentences that were generally hard to translate, since these were of no interest for comparing their system with others.

In order to measure translatability of a document, however, this information from Bohan et al. (2000) could be useful. One approach could be to investigate if these generally hard constructions exist in the document, and if so, how frequently.

3.9.3 The BLEU method

Until the year 2001, evaluation has not been a powerful tool in machine translation research, since no well-known automatic techniques had been developed for this (Doddington, 2002). This changed when IBM introduced BLEU. BLEU is short for “the bilingual evaluation understudy” and is a method of comparing a machine translation to a number of reference translations (Doddington, 2002).

The BLEU method is simple and elegant, and it is easy to see the connections to the concepts of precision and recall; see Section 8.5 for a discussion of these concepts. In order for a candidate translation to be close to reference translations, it should contain about the same words in about the same sequence and nothing more, nothing less. This can be viewed as four requirements, which BLEU takes care of in the following way (Papineni et al., 2001):

- All words in the candidate translation are searched for in the reference translations. BLEU counts the percentage of these words that are found. This awards using approved words.

- All word groups of size up to a certain number of words; i.e., N-grams up to a certain N, in the candidate translation are searched for in the reference translations and counted in the same way. This awards placing the words in the right order.

- Extra words in the candidate translation will lower the score since they reduce the percentages mentioned above.

- To handle the case where the candidate translation is too brief, a penalty to the score is subtracted when the candidate is shorter than all of the reference translations.
The first three cases are related to precision while the fourth is related to recall.

The Bleu metric ranges from 0 to 1 and the exact formula is (Papinemi et al., 2001):

\[
\text{Bleu} = \text{BP} \cdot \exp \left( \sum_{n=1}^{N} w_n \log p_n \right)
\]

Here, \(p_n\) is the percentage of word combinations of length \(n\) in the candidate translation that appear in at least one of the reference translations\(^4\). Further, \(w_n\) are weights; i.e., they decide for instance if the two-word combinations are more important than the three-word combinations. The constant \(N\) is the largest combination of words to consider.

Standard values are \(w_n = 1/N\) and \(N = 4\), which essentially means that the formula becomes a geometrical mean of all the occurrence percentages of N-grams up to the length of 4.

\(\text{BP}\) is the penalty for short translations, called brevity penalty. It is calculated as follows:

\[
\text{BP} = \begin{cases} 
1 & \text{if } c > r \\
 \frac{e^{1-r/c}}{e^{1-r}} & \text{if } c \leq r
\end{cases}
\]

The parameters \(c\) and \(r\) denote the candidate translation length and the effective reference corpus length\(^5\) respectively.

It is important to point out that Bleu is not designed for testing individual sentences, but for entire documents (Doddington, 2002).

**Evaluation of the Blue measure**

The N-gram technique that Blue is based on is very promising, according to Doddington (2002). There is no doubt that an automated evaluation method will allow for more new ideas in machine translation, since the traditional extensive and expensive human evaluation can be removed or postponed in the development process (Papinemi et al., 2001).

\(^4\)If the same word appears more times in the candidate translation than in every reference translation, the excess occurrences are not counted as found.

\(^5\)The effective reference length is calculated “by summing the best match lengths for each candidate sentence in the corpus” (Papinemi et al., 2001, p.5). Doddington (2002) treats this as the length of the shortest reference translation.
However, one important question remains—how well does the Blue score compare to human ratings of translation quality, with respect to adequacy, fidelity, and fluency of the translation.

Papinemi et al. (2001) states that the search for single word occurrence captures the adequacy aspect, while the word combination search accounts for fluency. They also correlated Blue scores to human ratings of machine translations and human translations. The study involved two groups of human raters—monolingual as well as bilingual. Their result was very promising and they write:

Given Bleu's sensitivity in distinguishing small differences between systems, we anticipate that machine translation research is at the dawn of a new era in which significant progress occurs because researchers can more easily home in on effective modeling ideas. (Papinemi et al., 2001, p.9)

An important remark is that while the Bleu score is useful to select the best translation out of several candidates, it must be seen as a relative and not an absolute grade of the translations. A reason for this is that if the number of reference translations increases, the Bleu scores will generally become higher (Papinemi et al., 2001).

Enhancing the Bleu measure

The invention of the Bleu measure was so successful that DARPA\(^6\) commissioned NIST\(^7\) to continue the work and develop a machine translation evaluation facility (Doddington, 2002). In this processes, the NIST MEASURE was founded.

The Nist measure is based on the Bleu measure, with the following important changes:

- Nist uses an arithmetic mean instead of geometric mean of the N-gram counts.
- Nist weights informative N-grams more heavily; i.e., words and word combinations that are not common are more important to investigate.
- Nist have a non-linear brevity penalty, which is milder when the candidate translation is close to the reference translations, but harder when significantly shorter.

\(^6\)the Defense Advanced Research Project Agency, which is part of the USA’s Department of Defense

\(^7\)the National Institute of Standards and Technology in the USA
Doddington (2002) have shown significantly better adequacy results with the Nist measure compared to the Blue measure, while not discovering any improvements on fluency results. In addition, it seems like the Nist score is less affected when adding more reference translations. Doddington (2002) writes that there appears to be no significant improvement in the correlation with human judgments with the use of more than one reference translation.

In Blue as well as in Nist, punctuation characters are treated as words (Papinemi et al., 2001). The question of preserving or removing case have been discussed, but an experiment by Doddington (2002) suggests the difference in results are very small.

Other suggestions for improving Bleu has been made by, for instance Hovy and Ravichandran (2003), who have shown interesting results by combining the original Bleu score with another round of Blue measuring after all words have been substituted for a marker of their parts-of-speech. This makes sure to reward sentences of the same grammatical structure as in a reference translation.
Chapter 4

Style guides for writing

In this chapter we explain what a style guide is and the benefits gained from using a style guide during writing of software user documentation in English. Then we briefly introduce two of the most commonly used style guides for the software industry today—Microsoft Manual of Style for Technical Publications (MSTP) and The Chicago Manual of Style (CMS). We also briefly introduce a style guide that is a part of US governments effort to use Plain Language in all communication to the public. We end this chapter with a short description of common style issues that arise when writing technical documentation.

4.1 Definition of a style guide

Technical writers usually adopt style conventions which govern the content, organization, layout, grammar, and punctuation of the works published in their domains. The subject of writing style includes making choices about the resources of a language; e.g., combinations of words into phrases, clauses, sentences, paragraphs, sections, and chapters (Kirkman, 1992). Style is a term covering balance, tone, and emphasis in writing (Kirkman, 1992).

We have not been able to find an appropriate general definition of a style guide due to that the concept of a style guide is ambiguous and not standardized in the software domain. This is because style guides can be applied to many software development artifacts, according to Robbins (2004), such as user interfaces, source code, and web pages. A definition that fits well for this thesis is that a style guide is a tool that gives guidance on the use of language rules for precise languages intended for software application instructions and descriptions.
4.2 Benefits of a style guide

The purpose of a style guide is to assist technical writers in writing readably and consistently. Style guides are collections of rules for writers, and they dictate which form of language should be used to ensure that the language is used consistently. There are several reasons for keeping documentation of a text consistent. The first reason is that consistency improves the writer's efficiency and therefore decreases the cost to produce technical documentation (Benson, 1996). The second reason is that consistency makes it easier to maintain a project over time as different writers contribute to the revision of a project (Benson, 1996). The third reason is that consistency helps readers in the learning process since they can more easily build an accurate model of a system (Benson, 1996).

4.3 Well-known style guides

In this section, we take a look at the two of the most commonly used style guides for software documentation: Microsoft Manual of Style for Technical Publication (MSTP) and The Chicago Manual of Style (CMS).

4.3.1 Microsoft Manual of Style for Technical Publication

Microsoft Manual of Style for Technical Publication (MSTP) is intended to give advice about usage and spelling of general and computer-related terms, design and interface elements, sentence style, and technical writing issues (Microsoft Corporation, 1998). This style guide provides guidelines to technical writers, based on the experience and advice of writers and editors working across the Microsoft corporation (Microsoft Corporation, 1998). MSTP provides a lot of different information; i.e., advice on the correct spelling of a term, a review of how to write consistently, what to avoid and what to do instead, together with frequent examples of what is correct and incorrect (Microsoft Corporation, 1998).

4.3.2 The Chicago Manual of Style

The Chicago Manual of Style is a reference book used by many professionals working with print and non-print documents; e.g., authors, editors, and publishers (University of Chicago Press, 2003). Most of the rules in CMS are guidelines, not imperatives, according to University of Chicago Press (2003).
Its core target groups are writers and editors of textbooks and journals (University of Chicago Press, 2003). Guidelines given in CMS are based on the experience and advice of scholars, publishing professionals, and writers (University of Chicago Press, 2003).

In the introductory chapter of CMS, the editors list a great number of areas that CMS covers; e.g., book formats, typography, general points of style, and—in the most recent edition of CMS—grammar and usage (University of Chicago Press, 2003). In the grammar and usage chapter, outlines of the grammatical structure of English are given; e.g., how words and phrases should be put together to gain clarity and examples of the most common grammatical errors. In the same chapter, there is also a glossary of troublesome expressions and a list of words together with the correct prepositions to be used with them. Many publishing houses and universities use CMS as their principal reference (McArthur, 1998).

4.4 Plain Language

Defining plain language is, according to Mazur (2000), difficult. In her article, she writes:

Ask 10 people and you'll get 10 different answers. (Mazur, 2000, p.205)


[Plain language is] language that reflects the interests and needs of the reader and consumer rather than the legal, bureaucratic, or technological interests of the writer or of the organization that the writer represents. (Mazur, 2000, p.205)

Another definition is given by Martin Cutts, as cited in Mazur (2000). He is a research director of the Plain Language Commission in the United Kingdom and he defines plain language as follows:

[Plain language is] the writing and setting out of essential information in a way that gives a cooperative, motivated person a good chance of understanding the document at the first reading, and in the same sense that the writer meant it to be understood. (Mazur, 2000, p.205)
Steinberg’s definition refers more to the origin of the term plain language while Cutts’ definition suggests the broader goal of plain language involving both writing and setting out language so that the reader will understand (Mazur, 2000).

The modern plain language movement started in the United States and the United Kingdom. In 1978, the US president Jimmy Carter issued an executive order, requiring federal regulations to be written in plain English. This was revoked in 1981 by president Ronald Reagan. In 1998 president Bill Clinton issued a memorandum for the heads of executive departments and agencies demanding that all the writing of federal governments must be in plain language. According to Clinton (1998), plain language saves the government time, effort, and money.

Some reasons, according to Mazur (2000), why using plain language is effective are: readers understand documents better, readers prefer plain language, readers locate information faster, documents are easier to update and more cost-effective, and it is easier to train people.

Some top requirements of plain language, according to Kimble (2002) and PLAIN (2001), are:

- reader-focus
- cohesive structure\(^1\)
- active voice
- concise word use
- professional, jargon-free tone

In 2001, PLAIN (2004) produced a style guide for writing user-friendly documents. In the rest of this section we give a summary of the most interesting guidelines.

According to PLAIN (2001), the active voice is preferred when engaging the reader because it eliminates ambiguity about responsibilities; i.e., who is supposed to do what. To communicate effectively, according to PLAIN (2001), the strong majority of the sentences should be in the active voice. Kimble (2002) also recommends writers to prefer the active voice and to use the passive voice only if the agent is unknown or unimportant.

PLAIN (2001) and Kimble (2002) also recommend the use of short and medium-length sentences. As a guideline, Kimble (2002) recommends to

\(^1\) to give the what of message in the introduction, to provide information in logical order, to mention secondary information last
keep the average sentence length to about 20 words. Only one idea should be expressed in each sentence because it shows clear thinking (PLAIN, 2001).

Ambiguity can be reduced in several ways; e.g., by keeping subjects and objects close to their verbs, by putting the conditionals such as *only* or *always* and other modifiers next to the words they modify, and by putting long conditions after the main clause (PLAIN, 2001).

PLAIN (2001) recommends writers to use contractions when appropriate. Contractions help the reader by increasing reading speed and soften the tone in the documents produced (PLAIN, 2001).

Present tense should be used whenever possible (PLAIN, 2001). Using present tense clearly shows what is standard practice and causes less confusion for the reader than compound verbs (PLAIN, 2001). A document written in the present tense is more immediate and less complicated (PLAIN, 2001).

PLAIN (2001) also state that the writer should avoid noun clusters by using more prepositions and articles to clarify the relations among the words.

### 4.5 Style for technical writing

A common type of text written in the software domain is OPERATING INSTRUCTIONS, which tell people how to perform a process (Alley et al., 2004). Because processes are so common in engineering, engineers often write instructions. For example, an engineer may have to write specifications to technicians on how to write a software manual for computer users about running a certain program. If the software manual is disorganized, users of the program may waste valuable time searching for a command (Alley et al., 2004). Money, time, and sometimes health often depend on the quality of the writing in instructions.

#### 4.5.1 Instructions

Instructions should be written so that a process can be used without hesitation, accurately and economically (Kirkman, 1992). A text should fulfill the following requirements, according to Kirkman (1992):

- The exact meaning of each instruction must be clear
- The instructions must be comprehensible
- The instructions must be adequate; i.e., there must be enough information to permit operation or at least information about where additional information may be found
- The instructions must be complete; i.e., everything a procedure can do must be specified

- The steps must be in proper order

- The warnings must be well placed and clear

- The tone must not produce resentment with the user; instead, it must help get the wanted response

In documentation, a procedure is a short description of the steps the user takes to complete a specific task (Microsoft Corporation, 1998). In printed and on-line documentation, Microsoft Corporation (1998) recommends that procedures are set off from the main text by their formatting.

MSTP recommends that the technical writer should document the preferred method of performing a procedure if there is more than one way to do it. One way to present alternative methods is by writing special tips section.

The requirements above are only partly fulfilled by the consideration of style. Other important aspects to consider are the physical representation of information, the receiver's level of expertise, and the context in which the instructions are to be used (Kirkman, 1992).

### 4.5.2 Style for descriptive and explanatory writing

Most technical documentation consist of descriptions, explanations, and instructions (Kirkman, 1992). The style used for technical documents should be impersonal, but it does not mean that it should be dull, according to Kirkman (1992). Writing impersonally does not require the use of passive and roundabout style (Kirkman, 1992). Kirkman (1992) gives examples of explanatory writing that is readable and clean because of the cumulative benefits gained from writing directly and actively. Factors that concretely contribute to direct and active writing are, according to him:

- that the sentences are of different length, but on average short and not complex

- that the verbs are mainly active

- that the vocabulary is plain but accurate.
4.5.3 Sequences and warnings

According to Kirkman (1992), it is clumsy to group several instructions in one long paragraph. Instead, it is much clearer to write instructions as a sequence of short and imperative instructions that describe the order of events that actually occur in an activity. He also states that occasionally it is useful to use impersonal style to specify an agent using *must* or the infinitive form.

Microsoft Manual of Style for Technical Publication (MSTP) recommends that the writer always presents a procedure in a numbered list and in most cases introduce it with an infinitive phrase. Most procedures consist of a number of steps and MSTP recommends that the writer try to limit a procedure to seven or fewer steps. The reason for this, according to MSTP, is that instructional design experts say that seven is the maximum number of items that people can remember at once.

Alley et al. (2004) recommend that in the language for instructions, cautions should also be used to warn readers of difficult or dangerous steps. They also state that the correct placement of warnings is before dangerous steps, not after. Kirkman (1992) recommends that warnings should be clearly separated, not buried in the text.

According to Microsoft Corporation (1998), a warning—not a caution—should advise users that failure to take or avoid a specific action could result in physical harm to the user or the hardware.

4.5.4 Voice

Voice shows whether the subject acts (active voice) or is acted on (passive voice) (University of Chicago Press, 2003). Many writers whose first language is English tend to use the passive voice excessively and without any specific purpose in their texts (Rivero, 2004). However, excessive use of the passive voice in a text produces an undesirable effect: the prose becomes too heavy, difficult to read, and unnecessarily complicated (Rivero, 2004).

In MSTP it is recommended that the writer use the active voice in general and that the passive voice should be avoided except when necessary to avoid a wordy or awkward construction (Microsoft Corporation, 1998).

The CMS recommends the use of active voice in most cases because passive voice, as a matter of style, is typically inferior to active voice (University of Chicago Press, 2003).
4.5.5 Tense and mood

It is important to use TENSES and MOODS consistently when writing descriptions (Kirkman, 1992). It is otherwise difficult to keep clear the difference between features or possibilities that are hypothetical or possible, and those that do exist in the system.

MSTP recommends the use of simple present tense as much as possible and avoidance of all other tenses. Present tense helps readers scan the material quickly. Past and future tense are only to be used when it is confusing to use present tense.

The mood of verbs expresses the attitude of the speaker toward the subject (Microsoft Corporation, 1998). In English, there are three moods: the indicative mood, which makes general assertions, the imperative mood, which makes requests, and the subjunctive mood, which is becoming more uncommon and expresses hypothetical information, demands, or wishes; see Examples 4.1–4.3.

Example 4.1 Indicative: *If you click on the button, a dialog box appears.*

Example 4.2 Imperative: *Click on the button to see the dialog box.*

Example 4.3 Subjunctive: *For the dialog box to appear, it is necessary that the button be clicked.*

According to Microsoft Corporation (1998), the indicative mood should be used to express general information such as facts and explanations. The imperative mood should be limited to procedures and direct instructions (Microsoft Corporation, 1998). Since the subjunctive mood often expresses wishes, hypothesis, and conditions contrary to facts, MSTP recommends that this mood should be avoided as much as possible. The reason for this, according to Microsoft Corporation (1998), is that it is seldom used and that it is usually unnecessary in technical documentation. MSTP also recommends that the writer should refrain from shifting moods.

4.5.6 Contractions

According to Microsoft Corporation (1998), the use of contractions is at the discretion of the editor. An advantage of contractions are that they lend an informal tone and save space. A disadvantage is that it takes more time to decode a contraction than a complete phrase during reading (Microsoft Corporation, 1998).
4.5.7 Text functions

In 1934, Karl Bühler introduced the notion of text functions (Newmark, 1988a; Ingo, 1991). The three different functions, according to him, are EXPRESSIVE, INFORMATIVE, and VOCATIVE. The expressive function is centered on the author and his personal use of the language, while the informative function focuses on the content. The vocative function includes everything that the writer uses to affect the reader, mainly emotionally, in order to make sure that the reader gets the message (Newmark, 1988a).

Poems are mainly expressive, technical reports are heavily informative, and advertisements are typically vocative. However, Newmark (1988a) believes that no text has only one of these functions.

These text functions are especially important to consider when translating texts, since they require different levels of translation. An informative part should be translated as objective as possible with a referential view, while a more subjective view is necessary when translating the metaphors and allegories typical to an expressive text (Newmark, 1988a).

An even narrower classification based on Bühler's text functions is made by Ingo (1991). The three basic functions are divided further in a hierarchy of up to four levels; see Example 4.4–4.5.

Example 4.4 Level 1.1.1 (informative–storing–tabular): tables, diagrams, sketches

Example 4.5 Level 3.1.2.3 (imperative–guiding–affecting receivers behavior–recommending): advice, teaching material

This classification makes it easier for the translator to realize the differences in characteristics between texts of distinct categories in a given language. Knowledge of this kind is essential for the writer to produce pragmatically well-working translations (Ingo, 1991).

4.5.8 Syntactic cues

Kohl (1999) proposed guidelines for improving readability and translatability of individual sentences, clauses, and phrases by using syntactic cues.

SYNTACTIC CUES are language elements that help readers in analyzing sentence structures or to identify parts of speech, which helps sentence perception (Kohl, 1999). Some examples of syntactic cues in the English language are:

- articles such as a, an, and the
suffices

- prepositions such as by, for, with, in

- auxiliary verbs such as is/are/was/were, has/have/had, has been/have been/had been

- word order.

An example of a syntactic cue is when readers read “Once upon a...”; their knowledge of the structure of English helps them predict that the next word will be “time.” That prediction is confirmed when they see the letter ”t” at the beginning of the next word.

In English, there are cases when it is possible to omit syntactic cues. According to Kohl (1999), the historical emphasis on conciseness in technical writing leads many writers to deliberately eliminate syntactic cues from their documents. But writers should refrain from omitting syntactic cues because their benefits can be clearly demonstrated (Kohl, 1999).

Syntactic cues do not always improve a problematic sentence; however, they do often improve readability or translatability, according to Kohl (1999). Kohl (1999) states that the improvement is achieved in the following ways:

- by enabling readers, translators, and machine translation systems to analyze sentence structures more quickly and more correctly
- by making it easier to predict the structure of subsequent parts of a sentence, which can lead to increased reading speed
- by eliminating some types of ambiguities that may go unnoticed by translators, resulting in mistranslation, or may force translators to seek clarification or to make guesses.

The syntactic cue approach is supported by research from the fields of psycholinguistics and reading behavior (Kohl, 1999). The approach is accompanied by a heuristic procedure for learning to use syntactic cues effectively (Kohl, 1999). The heuristic procedure helps technical writers to identify potential ambiguities and readability problems in their writing by explaining where to look for syntactic clues and when to insert them if they are omitted (Kohl, 1999).
Part II

Controlled languages
Chapter 5

Introduction to controlled languages

There are various ways to control how a writer is allowed to express himself. Spelling according to a certain dictionary is one basic constraint. Grammatical rules are a bit more complex since some rules are debated, and when writing poetry or advertisement texts, you can consciously break grammar rules to accomplish certain effects.

However, when writing technical documents where many writers contribute, it is important to have a common style in order to make the texts consistent, as mentioned in Chapter 4.

The most important goal when writing instructions and documentation must be that there should be no doubt to the reader as to what you mean, even if this comes at the expense of the variation and beauty of the texts. George Orwell (1946) wrote the following lines about language in politics:

> If you simplify your English, you are freed from the worst follies of orthodoxy. You cannot speak any of the necessary dialects, and when you make a stupid remark its stupidity will be obvious, even to yourself. (Orwell, 1946)

Just as in politics, there is a tendency among technical writers to develop special vocabularies, styles, and grammatical constructions. The result is texts that possibly are grammatically correct, but still troublesome to read, even for experts (Wojcik and Hoard, 1996).

There is a need to limit the vast variation possibility of natural language. The solution is called CONTROLLED LANGUAGES. In his article, van der Eijk (1998) writes:
Controlled Languages (CLs) are precisely defined subsets or variants of language constructed for use in particular environments for specific purposes. They are used chiefly for the authoring of technical documentation, and to support translation. (van der Eijk, 1998, p.4)

5.1 Controlled language components

The two fundamental principles of writing user documentation, according to Ericsson (1983), are that:

- the meaning must be clear
- the language must be simple.

This corresponds well to the main ingredients in a controlled language, namely:

- a vocabulary
- a set of grammatical rules.

In other words, a controlled language is built from the same parts as a natural language. The difference is that the vocabulary of controlled languages is limited and that the grammatical rules are well formalized (Altwarg, 2000).

In order to follow the first principle from Ericsson (1983), the vocabulary must consist of words that are easy to understand and have a well defined meaning. For instance, a written word should correspond as much as possible to a specific thought or concept. This is discussed in detail in Chapter 6.

The second principle governs the choice of grammatical rules. Some grammatical constructions, such as passive phrases, are harder to read and comprehend (Gunnarsson, 1989). They should therefore be avoided when maximum clarity is necessary.

5.2 Sublanguages

A similar term to controlled languages, namely SUBLANGUAGES, is also used in linguistics. A sublanguage is also a subset, or a part, of a language. More specifically, a sublanguage is the language used in a specific domain, such as in politics. However, sublanguages have evolved naturally while the terms, syntax, and semantics of controlled languages are purposefully limited (Altwarg, 2000).
Sometimes, the term *sublanguage* is used more loosely, meaning the domain specific language used in a particular type of text; e.g., installation manuals or support articles (Arnold et al., 1994).

### 5.3 The first controlled language

Basic English is regarded as the first controlled language and was developed as early as in the 1930s by Charles K. Ogden (Nyberg et al., 2003). It was intended to be used both as an international language and as a foundation for learning English. It consisted of the vocabulary together with a few writing rules, which described inflection and derivation of the words. However, Basic English, was never suitable for any practical purpose and therefore never used in the industry (Nyberg et al., 2003). The first industrial controlled language is described in Chapter 7.

### 5.4 Comprehensibility and readability of controlled languages

Controlled languages intend to improve text comprehension for both humans and computers (Nyberg et al., 2003). The factors that affect comprehensibility and readability in a negative way mentioned in Section 2.3 are eliminated by introduction of a controlled language, according to Nyberg et al. (2003).

The result of the use of controlled languages is that all documents will exhibit uniformity in word choice, use of terminology, sentence structure, and style (Nyberg et al., 2003). The uniformity makes the documents easy to maintain and reuse, Nyberg et al. (2003) explain. Another result is that the reusability of a source text increases dramatically, which reduces the overall cost of authoring new documentation (Nyberg et al., 2003).

ACEMA Simplified English\(^1\) (SE) is considered one of the most comprehensive examples of controlled English in use today (Shubert et al., 1995). In 1995, Shubert et al. (1995) conducted an experiment in order to determine whether procedure descriptions written in SE would be more effective than the same procedure descriptions written in non-SE. The results showed that SE significantly improves the comprehensibility of complex documents and makes it easier to find the location of the answers. This applies to both native and non-native speakers of English (Shubert et al., 1995).

\(^1\) ACEMA Simplified English is a controlled language described in Section 7.4
5.5 Translatability of controlled languages

Controlled languages have been shown to significantly ease the translation process of technical documentation (Adriaens and Schreurs, 1992). A common problem with translating technical documents is that the translators have little knowledge of the domain and therefore have trouble in, for instance, choosing the correct interpretation of an unusual ambiguous word. That, in turn, increases the risk of choosing the wrong translation. Controlled languages have been developed with this in mind, and to counter the tendency of writers to use inconsistent language (Wojcik and Hoard, 1996).

The importance of the translator’s understanding of the source text has been discussed in Section 3.2. The following words by van der Eijk (1998) emphasize this:

[I]t should be clear that the most obvious benefit that CLs bring to technical translation is that comprehensibility of source documentation is as beneficial to human translators as to any other reader. It prevents misinterpretations (which obviously may have dramatic consequences) and can actually help translators make significant time savings. (van der Eijk, 1998, p.4)

When using machine translation (MT), controlled languages have been shown to significantly reduce the post-editing effort, and their use is therefore encouraged by major vendors of MT software (van der Eijk, 1998). Vendors of translation memory products as well as terminology management tools have reported the same benefits of using controlled languages (van der Eijk, 1998). The reason is probably that the lexical and syntactic standardization that controlled languages bring increases the applicability of these systems, which in turn makes the overall human translation more efficient (van der Eijk, 1998).

As mentioned in Section 3.9.1, Spyridakis et al. (1997) invented a method for measuring translatability. This method was carried out on texts written in AECMA Simplified English\(^2\) (SE). They concluded:

[I]n certain cases using SE as a source language improves the quality and ease of translations. (Spyridakis et al., 1997, p.10)

This improvement seems to be welcome. Roukens (1998) writes:

---

\(^2\)AECMA Simplified English is a controlled language described in Section 7.4
CHAPTER 5. INTRODUCTION TO CONTROLLED LANGUAGES

In the 21st century [...] quality of translation will be an important determining factor for the quality of life. Economically speaking, translation will be big business, allowing ideas and facts generated anywhere on the globe to be spread everywhere. Any method or tool that will reduce the effort involved in this while improving the quality of translation will be welcomed by the industry. (Roukens, 1998, p.1)

In their study, Spyridakis et al. (1997) found that texts translated to Spanish received significantly better ratings when the source was written according to Simplified English rather than non-SE English. The same effect, however, was not measured for translations to Chinese. Spyridakis et al. (1997) suggest that the reason for this could be the greater linguistic similarity between Spanish and English than between Chinese and English.

Spyridakis et al. (1997) finish their article with the following recommendation:

[W]e can suggest some ways in which a company whose documents are frequently translated might use our results. Such a company should definitely investigate using SE or a similar controlled language as a standard for authoring procedural documents that are to be translated [...] (Spyridakis et al., 1997, p.11)

Reuther (2003) studied the question of how rules for readability and translatability relate. Her results showed that most controlled language rules improve both of these qualities. She did not find any rules that improved readability but not translatability; thus readability rules can be regarded as a subset of translatability rules (Reuther, 2003).

5.6 Human-oriented and machine-oriented CL

Nygberg et al. (2003) discuss the difference between controlled languages (CLs) aimed at improving comprehension by humans and those which are intended to simplify text processing by machines. The two categories have much in common, since many simplifications improve both of these aspects. For instance, limiting sentence length and requiring commas between conjoined sentences is beneficial to both humans and machines (Nygberg et al., 2003).

As mentioned in Section 3.5.4, anaphora is especially difficult for machines, which makes it a good idea to restrict the use of pronouns in machine-oriented controlled languages.
When human comprehension is important, word ordering must be considered carefully. A machine is able to process texts in various ways, while humans prefer reading from left to right\(^3\).

**Example 5.1** Place dependent clauses first when they express conditions on the action of the main clause.

**Example 5.2** Make your instructions as specific as possible.

Rules such as Example 5.1 are helpful to humans, but do not make the processing easier for machines (Nygberg et al., 2003).

Finally, CLs developed with machine processing in mind have stricter rules than human-oriented CLs. Rules such as Example 5.2 are too vague for computers to handle (Nygberg et al., 2003). However, Nyberg et al. (2003) conclude that it is often difficult to classify a CL as either human-oriented and machine-oriented, since the simplifications works both ways.

### 5.7 Drawbacks of CLs

There are, naturally, a number of drawbacks of controlled languages (CLs). Without doubt, introducing a controlled language requires time and money, just like any new introduction to a project. Here we will focus on more long-term effects, such as resulting writing speed.

Nygberg et al. (2003) points out that unless a CL tool makes sure that the CL rules are followed, the author has to constantly be on his guard when writing, which slows down the process. Even if such a tool is available, the rewriting of unapproved sentences can take a considerable amount of time. Patrick Goyvaerts, as cited in Nyberg et al. (2003), claims that writing in a controlled language takes up to 20% longer. The extended revision phase in document production due to the introduction of controlled languages is, however, compensated by the reduced revision required at the editorial level (Nygberg et al., 2003).

Not surprisingly, the restrictions of CLs can make the authors feel that their power of expression is limited when their desired words are unapproved (Nygberg et al., 2003). Nyberg et al. (2003) suggest that the authors and translators should be involved in the language definition process, so their ideas can be used as input.

The question of whether texts written according to CL rules in some situations are worse than the original texts remains. Møller (2003) concludes

---

\(^3\)Even if, for instance, Middle Eastern languages are read from right to left, the idea of reading in a certain direction still applies.
that even though CL texts are preferred by non-experts, experts often prefer non-CL versions, although some of the CL rules are beneficial to both groups.

Nyberg et al. (2003) summarize their discussion of CL drawbacks with the following words:

Despite the additional costs of introducing CL into an existing document production process, the long-term advantages typically outweigh the costs for organizations which produce a high volume of documentation per year, and for who the gains in consistency, reusability, and translatability are highly significant. (Nyberg et al., 2003, p.250)
Chapter 6

Terminology management

The methods for processing terminology have evolved dramatically over the last years. In the past, keeping the word-stock for specialist groups was an onerous task, and few people had the ability or the dedication to pursue it systematically (Sager, 1990). Today, with the help of computers equipped with terminology management tools, almost anyone with knowledge of computers can carry out this work (Sager, 1990).

The result of this evolution is that terminology management has become more popular, with more terminologies produced and with a higher degree of usage (Sager, 1990). When the users of a term database can contribute to it, it is much more likely to be used than prescribed standards sent from a standardization institute; the result being better conforming documents (Sager, 1990).

As mentioned in Chapter 5, an important part of a controlled languages is the vocabulary. In this chapter, we will discuss such vocabularies and how they can be stored in term databases.

6.1 Terminology definitions

The word TERMINOLOGY would etymologically be interpreted as the science of terms1. However, according to Sager (1990), this interpretation is rejected by most terminologists. He instead gives the following three definitions:

1. the set of practices and methods used for the collection, description and presentation of terms

2. a theory; i.e., the set of premises, arguments and conclusions required for explaining the relationships between concepts and terms

1It can also be interpreted as the study of terms, or the knowledge of terms.
The definition of a term seems to be easy. Sager (1990) writes:

Terms are the linguistic representation of concepts. (Sager, 1990, p.57)

Unfortunately, we have to define a concept in order to understand this, which is a much harder task. Sager (1990) gives a few definitions and later attempts to escape the problem by leaving it undefined as an axiom. Some of the suggestions are:

- Concepts are mental constructs, abstractions which may be used in classifying the individual objects of the inner and outer world. (British Standard Recommendation for the selection, formation and definition of technical terms, BS.3669:1963, as cited in Sager, 1990)

- A concept is a unit of thought, produced by grouping of individual objects related by common characteristics. (Draft of a German DIN document, as cited in Sager, 1990)

It is evident that concepts exist in our minds and that terms are the words that describe these concepts. In order to explain the connection between a term and a concept, we use a definition (Sager, 1990). Terms, concepts and definitions are thus the building blocks of terminology studies.

There is actually a simpler definition of the term term. In the language of a specific domain or discipline; i.e., in a particular sublanguage, the items which are characterized by special reference are the terms of that discipline, and together they form its terminology (in the third sense of the word) (Sager, 1990). Sager (1990) writes that the opposite of terms—items of a general kind, with references across a variety of sublanguages—is called words, and that their totality is called a vocabulary.

This naming is rather unlucky, since vocabulary often refers to a collection of words that sometimes are terms and sometimes not. In this thesis, we will use general words and general vocabulary to specify words and collections of words that are not terms.

6.2 Functions of terminology

According to Sager (1990), the subject of terminology has two different functions. The main function is to improve communication; that improvement is
then the only measure of success. The other function is to provide a record of a special sublanguage. This can then serve as advisory services for documentation and translation, or it can be published as a dictionary (Sager, 1990).

Advances in natural language processing have made way for another application of term databases. Spelling-checkers, expert systems and different office automation tools all benefit from term databases (Sager, 1990). Sager (1990) especially points out that developers of machine translation systems should consider using term databases, where term information is stored centrally, instead of using multiple dictionaries for analysis, transfer, and synthesis respectively.

### 6.3 One concept, one term

The idea that each and every word should mean the same thing for all people in all situations is not realistic. However, in a delicate situation in an exact science, it is vital that there are no misunderstandings. For instance, a surgeon calls for each tool by its exact name, and the term scalpel is defined in such a way that the surgeon and the nurse share the same conceptual view of what it refers to. The key is that within a specific domain, a term should be unambiguously defined (Sager, 1990).

Often, lexical items do double duty; an everyday word can be a strict term in a specific domain, as is 'noise' in communication science (Sager, 1990). The more frequent a term becomes, the more probable it is that its name will acquire figurative senses (Newmark, 1988b).

An effect of term standardization is that there is not the same need for emphasizing different attributes as there is of common words (Sager, 1990). We speak of strong wind, but a meteorologist would not say strong gale force 9, since gale force 9 is specific enough and effectively removes any thoughts of other attributes of wind than is intended (Sager, 1990).

A problem with this exact mapping between a concept and a term is that it is only useful if shared by a group of people, and standardizing the concept by means of language is impossible without the use of other terms (Sager, 1990). Therefore, Sager (1990) distinguishes two types of standards: those which has been made through direct empirical observation and those which are only described by language. He calls the categories PURE STANDARDS and PSEUDO-STANDARDS respectively.

Another interesting fact is that one term can be written in various ways. One common type of variation to a term is abbreviation. Sager (1990) summarizes a study by Hope (1984) concerning abbreviation in special-language
compound terms.

Example 6.1  dynamo strap clamp bolt $\Rightarrow$ dynamo clamp bolt $\Rightarrow$ clamp bolt

Example 6.1 is taken from this study by Hope, and shows the shortened form "dynamo clamp bolt". These shortened forms are mostly found in situation-conditioned texts, where the context makes ambiguity impossible (Sager, 1990). Sometimes it is difficult to determine if what looks like an abbreviation actually is another, more general term, used synonymously for the narrower term (Sager, 1990). This problem is evident from the last form of Example 6.1: "clamp bolt".

6.4 Structuring concepts

In a normal dictionary, uncommon words are often explained by means of other words, which the user is assumed to know. Since many words are polysemous and overlap in meaning, they are often defined by listing of synonyms. Few synonyms, if any, mean exactly the same, but if several synonyms are given, the meaning of a word can be interpreted as the common features of these synonyms (Sager, 1990).

With terms, the structure is different. A concept cannot be defined by synonyms of a term if we believe in the one-to-one correspondence between terms and concepts. However, it can be defined by the surrounding concepts in the domain (Sager, 1990). A gale, for instance is a wind that is stronger than a breeze but lighter than a storm; thus, the term gale could be found between breeze and storm in the field of meteorological terms, without any gaps or overlap.

This systematization of terms, and of concepts, are carried out very clearly in domains such as chemistry and biology. Sager (1990) writes that it does not exist a generally accepted or acceptable classification of concepts beyond these well established domains. In other areas, it is harder to describe the term relations even though we are aware of them.

6.5 Term collection

There are many ways to collect terms for a term database. Since new terms are invented each day, it is important to have routines for adding terms.

A terminologist identifying terms in a text would first scan the text and
split it into lexical units\(^2\), which have to be confirmed as to their terminological status. If the lexical unit precisely and uniquely names a particular entity, process or property of the domain in question, it can be declared a term (Sager, 1990).

This method of starting from words and looking for their meaning is called a SEMASIOLOGICAL approach.

An early example of a term database was Lexis, which helped translators in translating technical texts (Hoffmann, 1987). Lexis mainly consisted of compounds used in different special fields such as natural science, technology, and medicine. Each entry in the Lexis database consisted of the term in German and in one of several other European languages. In addition to this, a subject field code was stored as well as quality symbols and a short comment (Hoffmann, 1987).

Since Lexis was meant as help for translators, the translators' needs were controlling the addition of terms. The translator would send a list of terms that he would like to receive translated, and when he got the answer from the system, he could send back comments and suggestions for modifications and additions (Hoffmann, 1987). These suggestions would then be handled by the editors. Thus, the editors are updating the database with the help of the translators (Hoffmann, 1987).

Sager (1990) also mentions the benefits of logging unsuccessful queries to term databases for updating purposes. Since these systems are accessed directly through a computer, it is an easy task (Sager, 1990).

### 6.6 Term creation

Sometimes it is necessary to create a new term. Sager (1990) gives a list of highly idealized requirements, which is given here in a somewhat shortened form:

1. The term must relate directly to the concept. It must express the concept clearly.

2. The term must be lexically systematic. It must follow an existing lexical pattern.

3. The term must conform to the general rules of word-formation of the language.

\(^2\)Sager (1990) divides these further into uniterms, compound terms, and terminological syntagmas
4. The term should be capable of providing derivatives.

5. Terms should not have no redundant repetition (such as “IT Technology”).

6. Without sacrificing precision, terms should be concise and not contain unnecessary information.

7. There should be no synonyms, whether absolute, relative or apparent.

8. Terms should not have morphological variants.

9. Terms should not have homonyms.

10. Terms should be monosemic.

11. The content of terms should be precise and not overlap in meaning with other terms.

12. The meaning of the term should be independent of context. (Sager, 1990, pp.89-90)

The method of constructing terms from concepts is a matter of naming, and this is sometimes referred to as the ONOMASIOLOGICAL approach.

Sager (1990) stresses that it is important to teach sound naming conventions to scientists, technologists and translators, since it is likely that they are the ones who will invent new terms, because it is hard for terminologists to change a term that has started to get acceptance (Sager, 1990).

6.7 Term classification

In order to reflect that a term is connected to a concept, it is important that information about this concept is found in the term database. Subject code is, according to Nedobity (1987) one of the most important data elements in these databases. He writes:

A concept can only be defined if it is known which subject field, i.e. which system of concepts it belongs to. (Nedobity, 1987)

These subject codes help in differentiating homonyms and they limit the scope where a concept is valid (Nedobity, 1987). Other benefits of this classification is that it is easy to search the database for terms related to a specific term within the particular subject field (Nedobity, 1987).
If subject codes are standardized, terms can be exchanged between databases, and entire databases can be merged without having to produce concordances (Nedobity, 1987).

The subject field is hence a necessity for a term or concept to be defined, but the question of the definition itself remains. There are actually different kinds of definitions, fulfilling different needs. Sager (1990) lists three types of definitions:

- terminological definitions, which serve to single out the essential characteristics of the intension and to delineate the extension by reference to other terms
- intensional definitions, which are used by subject specialists for determining the precise reference of a term
- encyclopedic definitions, which aim to give non-specialists some degree of understanding a term

When the terminologist compiles a terminology, he needs an external definition. This definition does not necessarily fit the requirements of for instance precision, and in these cases the terminologist has to construct his own terminological definition (Sager, 1990).

The subject specialists know the concept and only need to be reminded of the terminological structure. They will thus be satisfied with an intensional or a terminological definition. The non-specialists, on the other hand, would probably benefit only from an encyclopedic definition.

Another large group of term database users are translators. They sometimes need, apart from the translation of the term, a definition of the target language equivalent together with a subject code and possibly a usage note (Sager, 1990).

Depending on the intended usage of the term database, not all terms have to be defined. Many terms are clearly understood and require no definition at all. If some term definitions can be collected from another authoritative database, there is naturally not a need to redefine these terms, as doing so may lead to misunderstandings (Sager, 1990).

6.8 Term translation

A bilingual term database is just as important as a standard dictionary to a translator within a specific domain. It is important to realize that a matching target language term only can be viewed as a suitable TARGET LANGUAGE
EQUIVALENT and not the linguistic representation of a matching concept (Sager, 1990).

From a translator’s view, it is important that a term is well defined in the term database. While a subject specialist uses term databases mostly for verification, a translator or another non-specialist; e.g., a communicator, needs it to understand the concept and possibly to paraphrase it (Sager, 1990).

Sager (1990) writes that large organizations are starting to realize their need to support their communication services with term databases.

6.9 Term database design

When discussing design of term databases, Sager (1990) talks about a three-leveled architecture, which he describes as follows:

- The bottom layer consists of collections of raw data, such as files defining terms and showing them in context.
- The top layer consists of complete products such as term banks, dictionaries, and translation aids.
- Between these layers is the middle layer, where the actual information is stored for each term, and this is often viewed as the database itself. The information in these fields are often linked to the bottom layer, giving access to the actual source of the information, such as the context of the definition (Sager, 1990).

The reason for keeping the bottom layer is that the user can decide if a term is appropriate in a given context by using this source data together with a usage note or other pragmatic data (Sager, 1990). This removes the necessity of the quality markers used in earlier designs (Sager, 1990).

6.9.1 General term fields

There are various information elements that can be stored for each term in a term database. Hoffmann (1987) wrote a recommendation based on her experience with Lexis. The following elements should, according to her, be sufficient:

- source language term/target language term with a minimum of grammatical information (if any)
• definition and/or context
• subject field, source, quality
• status or authority (e.g., deprecated, obsolete)
• date of entry, date of last change

Sager (1990) gives an example of information fields in a typical term database. This information is summarized in Table 6.1.

Table 6.1: Information fields in a term database (Sager, 1990, p.144)

<table>
<thead>
<tr>
<th>CORPORA OF RAW DATA</th>
<th>SOURCE INFORMATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conceptual data</td>
<td>Lingustic data</td>
</tr>
<tr>
<td>language</td>
<td>language</td>
</tr>
<tr>
<td>definition</td>
<td>term</td>
</tr>
<tr>
<td>grammatical information</td>
<td>context</td>
</tr>
<tr>
<td>links to other concepts</td>
<td>synonyms</td>
</tr>
<tr>
<td>scope notes</td>
<td>abbreviation</td>
</tr>
<tr>
<td>subject field</td>
<td>variants</td>
</tr>
<tr>
<td>ADMINISTRATIVE DATA</td>
<td>e.g. record number, date</td>
</tr>
<tr>
<td></td>
<td>of creation/update,</td>
</tr>
<tr>
<td></td>
<td>terminologist</td>
</tr>
<tr>
<td>Foreign lang. equiv. data</td>
<td>grammatical information</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In a term database, depending on the intended usage, it can be beneficial to store links between concepts. This can be of simple kinds as hyperonymy and hyponymy; i.e., broader or narrower terms, or regard other relations such as when one term is a property of another term (Sager, 1990). This information is of great help to a user who is not quite satisfied with the term he found (Sager, 1990). He can then search for other similar terms by following these cross-references.

Newmark (1988a) stresses that the term database becomes much more useful for writers and translators if it includes information such as frequency and formality of the terms, and if the terms are divided in lexical fields.

6.9.2 Fields for the translator

There are many information fields about a term that are helpful to a translator. Sager (1990) has graduated the importance as:
6.9.3 Grammatical fields

An interesting example of grammatical information is which prepositions that are likely to be used with a verb. Collins Cobuild English Language Dictionary is a database with that information stored. A sample, taken from Hindle and Rooth (1991), is presented in Table 6.2.

<table>
<thead>
<tr>
<th>verb</th>
<th>prepositions</th>
</tr>
</thead>
<tbody>
<tr>
<td>appropriate</td>
<td>for</td>
</tr>
<tr>
<td>ask</td>
<td>about</td>
</tr>
<tr>
<td>assign</td>
<td>to</td>
</tr>
<tr>
<td>assist</td>
<td>in, with</td>
</tr>
</tbody>
</table>

University of Chicago Press (2003) also has a section dedicated to related prepositions in The Chicago Manual of Style, which includes other parts-of-speech than verbs.

Apart from helping the writer to choose the correct constructions, this kind of information is most valuable when resolving anaphora (see Section 3.5.4) because it can suggest the correct prepositional phrase attachment.

6.10 Controlled language vocabularies

In this chapter we have discussed central parts of terminology science, such as the creation, collection and classification of terms. When introducing controlled languages, an important part is the limitation of terms.

The one-to-one correspondence between words and concepts is a common goal of controlled languages (Nyberg et al., 2003).
The complicated procedure of selecting the appropriate terms for a controlled language is outside the scope of this thesis. However, assuming that such a collection is produced, there is an important addition to the term database that should be mentioned: A typical controlled language term database should include a field designating if the term is desired or non-desired in the controlled language (Heinämäki, 1997; Nyberg et al., 2003). If the term is non-desired, a reference to a better term or another way of writing is helpful to the user. Table 6.3 and 6.4 shows how this information is available in two different controlled language term databases.

Table 6.3: Example from the Ericsson English vocabulary (Ericsson, 1983)

<table>
<thead>
<tr>
<th>term</th>
<th>usage or preferred alternative</th>
</tr>
</thead>
<tbody>
<tr>
<td>including (prep)</td>
<td>comprising; taking account of</td>
</tr>
<tr>
<td>incoming (adj)</td>
<td>arriving</td>
</tr>
<tr>
<td>incomplete (adj)</td>
<td>USE not complete</td>
</tr>
</tbody>
</table>

Table 6.4: Example from AECMA SE vocabulary (AECMA, 2004)

<table>
<thead>
<tr>
<th>term</th>
<th>usage or preferred alternative</th>
<th>examples (approved and unapproved)</th>
</tr>
</thead>
<tbody>
<tr>
<td>few (adj)</td>
<td>USE some</td>
<td>Some bolts are shorter than others.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOT A few bolts are shorter than others.</td>
</tr>
<tr>
<td>filing (n)</td>
<td>USE particle</td>
<td>Remove all the metal particles.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOT Remove all the metal filings.</td>
</tr>
<tr>
<td>filtered (adj)</td>
<td>That “has gone” through a filter</td>
<td>Dry with low-pressure filtered air.</td>
</tr>
</tbody>
</table>
Chapter 7

Controlled language in industry

By now, there are hundreds of companies that have realized the benefits of controlled languages (CLs) and use them regularly. The primary reason for this is the resulting improvement in comprehensibility and translatability (Wojcik and Hoard, 1996).

Controlling language efficiently, in general, is a difficult problem, but when the domain is narrow the task becomes easier. That is why controlled languages are particularly effective in commercial or industrial applications such as the authoring of user manuals. There, the complexity of the documents is high and the volumes large; yet the terminology is domain specific (Altwarg, 2000).

The level of language control in industry ranges from informal company internal guidelines for technical writing to professional document authoring systems (van der Eijk, 1998). The latter sometimes include support for meta information and the use of full parsing to enforce validation of an application specific grammar (van der Eijk, 1998). Such tools are presented in Chapter 8. In this chapter we discuss a number of controlled languages developed for industrial use.

7.1 Caterpillar Fundamental English

The first industrial controlled language was designed in 1972 when Caterpillar Tractor Company introduced its Caterpillar Fundamental English (CFE) (Kamprath and Adolphson, 1998). Their goal was to eliminate the need to translate service manuals, which would be accomplished by writing in a limited language that could be learned by non-native speakers of English with only some basic training (Kamprath and Adolphson, 1998). Therefore, translation from CFE was not an issue, and CFE can be categorized as a
monolingual controlled language (Allen, 2004).

The allowed vocabulary of CFE consisted of as few as some 850 words (Allen, 1999b). CFE is a typical example of a limited vocabulary controlled language, where the most notable limitation upon the writer is the very few words allowed. There are limitations on style as well, but these tend to be vague and more like advice than rules (Allen, 1999b). This makes it hard to design supportive tools for the writer, other than simple checkers to ensure adherence to the vocabulary.

Even though Caterpillar abandoned its Caterpillar Fundamental English in 1982, it made way for several new controlled language projects (Kamprath and Adolphson, 1998). Among these are E. N. White's International Language of Service and Maintenance (ILSAM), the Plain English Program (PEP), which was invented by the firm Smart AI, and Perkins Approved Clear English (PACE) (Nyberg et al., 2003). PACE will be described in Section 7.3, and ILSAM can be considered the root of many of the other controlled languages described here, such as AECMA Simplified English, IBM's Easy English, and Ericsson English (Adriaens and Schreurs, 1992).

### 7.2 Caterpillar Technical English

Caterpillar did not give up on controlled languages after abandoning its Caterpillar Fundamental English. In 1991, they introduced Caterpillar Technical English (CTE), which was quite different from its predecessor (Kamprath and Adolphson, 1998). Caterpillar realized the need for translation and focused on translatability in CTE by introducing syntactic rules that could be mapped to about 10 other languages (Allen, 1999b). This classifies CTE as a multilingual controlled language (Allen, 2004).

The vocabulary initially consisted of 8,000 general words and 50,000 technical terms, and since then, new technical terms have been added together with their translations in a multilingual database. This approach is common to the group of extended vocabulary controlled languages (Allen, 1999b).

### 7.3 PACE

In the 1980s, the UK engineering company Perkins Engines International Limited introduced a controlled language called Perkins Approved Clear English (PACE) (Nyberg et al., 2003). The company is a leading world-wide manufacturer of engines. It has a large need for production of documenta-
tion in five languages because of the frequent introduction of new products and modification to existing products (Nyberg et al., 2003). PACE was designed to simplify English publications for non-native speakers, and to aid translation, both human and computer-assisted (Nyberg et al., 2003). This controlled language consists of a lexicon of approximately 2,500 words and 10 writing rules. The PACE writing rules are:

2. Omit redundant words.
3. Order the parts of the sentence logically.
4. Do not change constructions in mid-sentence.
5. Take care with the logic of “and” and “or”.
6. Avoid elliptical constructions.
7. Do not omit conjunctions or relatives.
8. Adhere to the PACE dictionary.
9. Avoid strings of nouns.
10. Do not use “ing” unless the word appears thus in the PACE dictionary. (Nyberg et al., 2003, p.257)

7.4 AECMA Simplified English

Perhaps the best known controlled language today is AECMA Simplified English, which is sometimes referred to just as Simplified English (SE). The language is the result of an investigation started in 1979 by the European Association of Aerospace Industries (AECMA) concerning readability of maintenance documentation in the civilian aircraft industry (Nyberg et al., 2003). A basic vocabulary and a collection of rules were set up, and in 1986, the first edition of “the SE Guide” was released (Nyberg et al., 2003).

SE is by far the most well-researched controlled language in practical use, and it has been shown to improve comprehensibility in a statistically significant way, according to van der Eijk (1998). He concludes a study as follows:
These results are encouraging and exciting: they show that com-
pliance to SE, and investment in SE implementations within an
organization, are not just a matter of meeting industry standards,
but are an investment for which an independent business case can
be made. (van der Eijk, 1998, p.5)

SE applies to two different types of text—what AECMA calls PROCEDURES
and DESCRIPTIVE WRITING. Procedures are mainly instructions, while de-
scriptive writing is of a more general informative kind (AECMA, 2004).

AECMA SE was originally intended only for the aircraft industry, but
SE has been used as an example when designing controlled languages within
other fields (Nyberg et al., 2003). The 57 grammar rules are possible to
transfer to other domains, but the vocabulary is more problematic, since
common words and domain specific terms are mixed.

7.5 Ericsson English

Ericsson English (EE) was created to help in making basic technical instruc-
tions and descriptions comprehensible to readers who have a poor command
of English (Ericsson, 1983). However, the project turned out to be a failure
because it was not sufficiently accepted by the employees (Hollqvist, as cited
in Spichtinger, 2000).

The most notable property of Ericsson English is its two-level design. All
Ericsson documents were classified as either “Level 1” or “Level 2”, where the
former required the most strict writing. Here is an excerpt that illustrates
this:

When you are writing Level 2 documents, you need not confine
your sentence constructions within the strict rules for Level 1;
but stay within those rules as much as you can, so that your
writing makes minimum demands on your readers’ command of
English. [...] Always look for the simplest possible forms of verbs
(preferably active forms) and keep sentence structures as simple
as possible. (Ericsson, 1983, "Instructions" p.3)

The vocabulary of EE is divided into the words allowed in all Ericsson docu-
ments and the words disallowed in Level 1 documents but accepted in Level 2
documents (Ericsson, 1983).

Another interesting aspect of EE is that it regulates the procedure when
there is a need for an unlisted word, which is shown in the following excerpt:
You must not add new words to the Word List unnecessarily. If you need to use a new word that is useful only in a very specialized context, give a definition of the word, in EE, in the document that you are writing. (Ericsson, 1983, "Instructions" p. 8)

7.6 Scania Swedish

Scania has developed the controlled language Scania Swedish, which is used by Scania’s technical writers when writing documentation (Granlund, 2002). These source texts are then translated by translation agencies to English, and from English to other languages (Granlund, 2002).

Scania Swedish was developed to ease translation and to standardize the terminology within Scania (Granlund, 2002). Scania Swedish primarily limits the following aspects of Swedish (Almqvist & Sågvall-Hein, as cited in Granlund, 2002):

- the number of synonyms
- the number of terms that are hard to understand or hard to translate
- the use of word stems for compounds and derivations

The concept of language control has helped Scania to uphold linguistic quality throughout the translation process (Granlund, 2002).

7.7 EasyEnglish

EasyEnglish is not described as a controlled language, but rather as a tool developed by IBM to help the writer produce clearer and simpler English (Bernt, 1997). A tool is clearly not a controlled language, but EasyEnglish governs how a text is allowed to be written and therefore ensures that some kind of controlled language is used.

In this section, we will write EASYENGLISH LANGUAGE when referring to the language ensured by the tool EasyEnglish.

Bernt (1997) claims that the EasyEnglish language is not a controlled language. His view seems to be that a controlled language is domain dependent and that its primary goal is to adapt texts to a particular machine translation engine. He also implies that controlled languages are built from rules governing which constructions are allowed in the language, while EasyEnglish clearly checks for what is not allowed. However, this definition is not universal and is not the one used in this thesis; see Section 5.1.
As Bernth (1997) writes, EasyEnglish is a grammar checker combined with extra layers of checks for ambiguity and complexity.

The reason for using EasyEnglish is, according to Bernth (1997), that documents become not only more adapted to machine translation, but easier to understand for native speakers as well as non-native speakers of English. He also states that their approach is more general than current controlled language solutions because it uses a broad-coverage English grammar to find more general types of ambiguity.

Some examples of ambiguities handled by EasyEnglish are:

- attachment of present participial clauses
- scope of coordinating conjunctions
- double passives

For more information about how the EasyEnglish tool works, see Section 8.3.4.

7.8 Controlled language interaction

There are reasons to question why there are no universally adopted controlled language within the technical domain. Some answers to that questions are given by van der Eijk (1998). First, the terminology is often specific to a typical subdomain or even to the company, product, and target user, which reduces the portability. Also, many companies see their controlled language as proprietary knowledge and experience which gives them a competitive edge, and therefore hold on to their information. Finally, the research community of natural language processing, according to van der Eijk (1998), seems to view the technical communication domain as uninteresting, instead of the rich source of research challenges that it is to those who are familiar with it.

7.9 Introducing controlled languages

When discussing introduction of controlled languages, Douglas (1996) mentions the CONFORMANCE PROBLEM. The problem is that no commonly used CL definition defines the language it covers in an exhaustive manner (Douglas, 1996). Douglas (1996) writes that even if there are a number of fairly specific rules, these rules are always additions to some vague substrate of good English, not formally defined. A result of this is that organizational
patterns of language use and feedback is equally important as the impoverished definitions (Douglas and Hurst, 1996).

In his study of controlled language introduction at KONE\(^1\), Loimaranta (2000) stresses the importance of a WRITER CERTIFICATION SYSTEM, in addition to a term bank, a CL checker\(^2\), and a translation memory. At KONE, the design of this system was similar to the design of a translation memory. Information about writer errors was recorded in a database, which allowed the quality of the writing to be quantified. Such information can then be used for initiating new authors and for customizing training to each writer’s needs (Loimaranta, 2000).

Douglas (1996) also points out that different document types can require different control language rules, even though the allowed vocabulary is the same. Douglas (1996) gives an example where procedural manuals and maintenance manuals for an aircraft may use the same terminology, but quite different constructions reflecting different functions of the text.

---

\(^1\)KONE is a Finnish global service and engineering company

\(^2\)Controlled language checkers will be discussed in Chapter 8.
Chapter 8

Controlled Language Tools

The primary tools for supporting authors who are using controlled languages are the controlled language checkers (CL checkers). A CL checker is a piece of software specialized for aiding the author in determining whether a text conforms to a particular CL (Nybørg et al., 2003).

In this chapter, we present some background on CL checkers and give examples of different CL checker used in industry. We will also mention another useful type of tool connected to controlled language—authoring memories. Another type of tool helps in managing term tabases. We will, however, not describe these, since this functionality to some extent is included in the CL checkers. We end this chapter with a discussion of some important quality measures with regard to controlled language tools.

8.1 Definition of CL checkers

There are a number of requirements that most controlled language support tools try to fulfill. This list is collected by van der Eijk (1998):

- Linguistic analysis of Controlled Language text
- Generation of useful critiques to authors
- General morpho-syntactic\(^1\) and spelling correction
- Support for interactive transformation of general sublanguage expressions into the controlled language
- Integration in standard DTP\(^2\) environments

---

\(^1\)Morpho-syntax is traditionally referred to as *grammar* (van Valin, 2001).
\(^2\)Desktop publishing
8.1.1 Red light/green light or interactive systems

There are two different kinds of tools for controlled language authoring. Today, most systems belong to what Allen (1999b) calls RED LIGHT/GREEN LIGHT SYSTEMS, where the author submits a complete text to the checker. Then the system gives comments on problematic sentences, one at a time.

The other approach is to assist the writer during the writing process. This kind of systems is called INTERACTIVE AUTHORING SYSTEMS (Allen, 1999b).

Simple substitutions can be performed automatically by both types of systems, such as changing all occurrences of a disallowed word to an allowed synonym. However, if words of other parts-of-speech are to be used instead of a disallowed word, the sentence must be restructured, which is harder for a computer. Resolving such situations often requires human judgement (Nyb erg et al., 2003).

8.1.2 Proscriptive or prescriptive approach

Writing rules can be described in disallowing form, stating which constructions are not allowed in the controlled language. Nyberg et al. (2003) call this the PROSCRIPTIVE APPROACH. The opposite kind of rules is those that describe the allowed constructions. This is analogously called the PRESCRIPTIVE APPROACH (Nyberg et al., 2003).

The proscriptive approach typically requires considerably less work, since these rules often can be implemented with simple pattern matching techniques (Nyberg et al., 2003). It is also possible to start with a single rule and extend the check with more disallowed constructions.

The prescriptive approach, on the other hand, requires that each and every linguistic structure allowed in the CL is defined; otherwise correct sentences will fail the test. However, this approach is less likely to give inappropriate feedback and overlook incorrect constructions (Nyberg et al., 2003).

8.2 Common problems for CL checkers

There is probably no surprise that adherence to some CL rules is harder to check than to others. Nyberg et al. (2003) illustrates this with the following three examples:

Example 8.1 Do not use sentences with more than 20 words.
Example 8.2 If possible, use an article before a noun phrase.

Example 8.3 Make your instructions as specific as possible.

While Example 8.1 is trivial to implement, 8.2 is harder, and 8.3 is impossible using current knowledge of computational linguistics (Nyberg et al., 2003).

8.3 CL checkers in industry

We will here give a sample of the successful controlled language tools used in the industry. The controlled languages supported by these checkers were described in Chapter 7, which we recommend reading first.

8.3.1 PACE Checker

The PACE Checker is an in-house checker developed by Perkins Engines Ltd. (Douglas and Hurst, 1996). The checker is based on a client-server architecture and has an interface in Microsoft Word.

The PACE checker is an interactive authoring system and any error found is marked up in the text with so called hot-spots (Douglas and Hurst, 1996). By double-clicking the hot spot, the user is able to launch a dialog box similar to the one found in the spellchecking dialog in Microsoft Word. The user can then revise and edit the text (Douglas and Hurst, 1996).

8.3.2 The Boeing checkers

Boeing has agreed to write all maintenance documentation using AECMA Simplified English. In order to support their authors, Boeing developed a CL checker called Boeing Simplified English Checker (BSEC) (Nyberg et al., 2003).

BSEC was developed for internal use only, and it was initially very simple. In order to solve syntactic ambiguities\textsuperscript{3}, BSEC used statistical methods (Nyberg et al., 2003).

Boeing later added semantic and pragmatic checking capabilities and called the new checker Enhanced Grammar, Style, and Content Checker (EFSC) (Nyberg et al., 2003).

\textsuperscript{3}see 3.5.2
8.3.3 Scania Checker

Scania Checker is an easy-to-use tool for checking compliance with Scania-Swedish; see Section 7.6. It is web-based and primarily designed for handling small texts; i.e., texts consisting of less than 5,000 words (Granlund, 2002).

Scania Checker is able to find three distinct types of errors in the source texts (Granlund, 2002):

- unknown words, which are not found in the ScaniaSwedish lexicon
- illegal words, which according to ScaniaSwedish should be substituted for other words
- grammatical errors, which requires a syntactic change.

8.3.4 EasyEnglish

EasyEnglish is an authoring tool from IBM, developed to help writers produce clearer and simpler English. As mentioned in Section 7.7, this is done by pointing out ambiguity and complexity in written texts, and by some standard grammar checking (Berth, 1997).

EasyEnglish is also equipped with a generation module, which makes suggestions for rephrasing, and a module to check words against a controlled vocabulary (Berth, 1997).

The vocabulary is divided into the following parts:

- words that should never be used
- words that should only be used as certain parts-of-speech
- acronyms that must be spelled out at least once before used

There is also an option to specify a user defined controlled vocabulary. In that case, EasyEnglish will give error reports on all encountered words that are not in that vocabulary (Berth, 1997).

A separate module can be used to identify candidates for terms in the text. These terms are intended for review and it is possible to, for instance, send the list of terms for translation before the actual text is complete in order to speed up the translation process (Berth, 1997).

Finally, EasyEnglish does not only make suggestions for modifications but does also give an overall clarity index of the scanned text (Berth, 1997), which can be used to identify especially problematic texts in a collection.
8.3.5 Maxit

The Maxit Checker is a software tool that analyzes technical texts written in either their own Controlled English or AECMA Simplified English\(^4\) (SMART Communications, Inc, 2005). It is offered commercially by SMART Communications, Inc. in New York, USA. The text is analyzed on basis of the following ten rules, according to Cremers (2003):

- write positive sentences
- write short sentences
- 1 though per sentence
- use approved terminology
- write simple sentences
- use active voice
- avoid gerunds and verbs used as nouns
- avoid conditional sentences
- avoid word clusters
- use valid abbreviations

The Maxit Checker is a plug-in for several authoring environments (Cremers, 2003). There are versions for Microsoft Word and FrameMaker\(^5\). The Maxit checker uses artificial intelligence, more than 8 500 English grammar rules, and dictionaries to electronically critique text, according to SMART Communications, Inc (2005). The critique is given as messages (Cremers, 2003), which for instance can look like: “awkward”, “change-to-noun”, “missing reference”, “not-in-dictionary”.

Some key features of Maxit are the custom dictionaries, the custom grammar rules and the color-coded Maxit messages. The writer selects a paragraph, page, or document for analysis and Maxit inserts color-coded messages to show the corrections. Maxit gives advice about usability, style, grammar, technical accuracy, consistency, product nomenclature, readability, liability, translatability, globalization, and cultural acceptance.

\(^4\)AECMA Simplified English is a controlled language described in Section 7.4

\(^5\)FrameMaker is a registered trademark of Adobe Corporation.
8.4 Authoring memories

During the past 10–20 years, many companies have used translation memories (TMs) for improving translation productivity (Allen, 1999a); see Section 3.7 for a deeper discussion of this. Recently, a new technology in the area of computer-assisted translation called **authoring memory** (AM) has been introduced.

The term *authoring memory* first appeared in Brockmann (1997)—an article about how a company that used a translation memory was considering the idea of adapting memory-based approaches to authoring and controlled language writing environments. Brockmann (1997) explains how a controlled language is a beneficial element to the use of authoring memories:

> The more controlled a source text, the more efficient these tools will be in the translation process. In the medium term, they will also be adopted for source-text authoring. This means that the writer will be able to re-use his or her own material using an “authoring memory”, thus increasing consistency even more in the source language. (Brockmann, 1997, p.2)

Controlled languages enhance the authoring memory because they provide an overall terminological consistency, which favours an AM-system, removes ambiguity and enables re-usability of linguistic structures (Allen, 1999a). With authoring memories, authors can ensure consistent use of terms and general words. Authoring memories also enables writers to search and retrieve what they have written in the past. The purpose of an authoring memory is to improve the computer-assisted translation process. Brockmann (1997) stated that:

> Controlled language, by definition, is characterized by consistent syntax and terminology. A translation memory tool, in combination with a powerful terminology database, can therefore be of valuable assistance in this context—helping to reduce translation cost, while at the same time making life easier for the translator. (Brockmann, 1997, p.1)

Allen (1999a) addresses a number of the positive and problematic issues for the development and use of translation memories, or authoring memories, in industrial environments. Allen (1999b) believes that authoring memories will improve technical writing environments just as much as translation memories have done for translation environments.
8.5 Measures

There are various ways to measure the efficiency of a CL tool. We will here describe four general measures that can be used for CL tools as for almost any other type of checking tool.

Precision

The term precision is a general measurement of how well a tool is able to select only those entries that it should select. A high precision score is equivalent to relatively few incorrect selections. However, the precision score does not say anything about how good the tool is at actually locating the entries that it is supposed to find.

\[
\text{precision} = \frac{\text{number of correct selections}}{\text{total number of selections}}
\]

In the domain of controlled language checkers, a high precision score means that the user is not often bothered with false error reports. According to Bernth (1997) it is the common view that user acceptance depends on suitably high precision. Still, Bernth (1997) writes that most grammar checkers seem to have a problem with precision and that this depends on the inability of the system to make sense of the input, which in turn is caused by ill-formed input in combination with too narrow coverage of the parser.

Recall

The term recall has the general meaning of the ability of a tool to select those entries that it is supposed to select. A high recall score means that the tool misses relatively few entries that should have been selected. The recall says nothing about if the tool also makes a lot of incorrect selections.

\[
\text{recall} = \frac{\text{number of correct selections}}{\text{number of entries that should be selected}}
\]

Recall and precision are thus two mathematically independent measurements that well complement each other.

In the controlled language domain, a high recall score means that the system to a high degree is able to find the violations made by the user. If the writer is observant of this and is able to make corrections, the result is a text that conforms well to the controlled language, or at least to the rules checked by the tool.
Usefulness

Precision and recall are two measurements of how good a tool is at doing what it is supposed to do. However, neither of them say anything about how useful that is for solving the particular problem behind. Therefore, it is important to analyze the result with respect to some parameters that match the problem; i.e., to measure the usefulness of the tool.

One general definition of usefulness, when speaking of controlled language tools, is “how well writers can use the system to arrive at a satisfactory document” (Bernth, 1997).

Naturally, there are different reasons for introducing controlled languages in different situations, so in order to decide what is satisfactory, one has to find measurements that match the reasons in question; see Chapter 2 and Chapter 3 for discussions on comprehensibility and translatability measures respectively.

One important fact is that since precision and recall measure how good a tool is at performing its task, it suffices to evaluate the usefulness of the task itself, since a combination of precision and recall measures the usefulness of the tool.

Convergence

CONVERGENCE is a measure appropriate for tools that attempt to automatically correct errors in the material. It shows how good the tool is at correcting these errors.

\[ \text{convergence} = \frac{\text{number of accepted corrections}}{\text{number of corrections made}} \]

A convergence of 100% means that the tool eliminates all errors that it attempts to correct and that it does not introduce any new errors in the process (Nyberg et al., 2003).
Part III
Case study
Chapter 9

Documentation at IFS

The user documentation for IFS Applications has two main purposes; one is to serve as a reference book and the other is to instruct and give understanding (Sjöqvist, 2005). At IFS the first purpose is fulfilled by window descriptions and the other by process descriptions. There are also other types of documentation text; e.g., about descriptions, activity descriptions, and process models.

In this chapter, we present some background on how IFS views user assistance and the current documentation process for IFS Applications. We also describe different types of documents produced at IFS and how they are related. Finally, we give a short description of the official style guide, IFS Style and Writing Guide, used at the company for ensuring consistency in the written material produced all over the company.

9.1 IFS’ view on user assistance

User documentation at IFS can be seen as a hierarchy of support levels for the user where he can find assistance when he tries to understand IFS applications (Hammarström, 2004a). The assistance for the user can be classified into five different levels; see Figure 9.1.

The first level of support is the application user interface. The second is assistance directly in the application; e.g., a so called wizard\(^1\). The third is the help documentation; e.g., a help pane in the application. The fourth is the user community, which consists of groups of people who share a common interest and communicate with each other about that interest through a web

\(^1\)A wizard is an interactive utility that leads a user through all of the steps required to complete a task.
The cost for the user documentation increases with the level in the hierarchy. If the user can get a good explanation of the concept that he wants to understand low in the user documentation hierarchy; i.e., closer to the user in the application, the expenses will be smaller both for the customer and IFS (Hammarström, 2004a). The most expensive kind of user support is having people working at IFS personally answering questions on the phone or through e-mail.

9.2 IFS' documentation structure

The IFS user documentation is process-oriented (Celinder, 2004). This means that the focus in the documentation is set on activities within the IFS Applications; i.e., what you can perform and accomplish in IFS Applications (Celinder, 2004). Therefore, most of the available document types are connected to the process models in one way or the other. IFS’ process models are a way to create an overview of an entire business and contain main processes that are supported by components in IFS Applications. The user can choose to explore the models from an industrial perspective or from a functional perspective. The first level provides an overview of the models. The user can reach different levels of the models by using the navigator or by clicking directly on the model. The activity diagrams, the most detailed level of the IFS process models, are connected to activity descriptions that provide details on how to carry out the activities.

The user documentation is divided into two logical parts—“online documentation standard” and “online documentation extended” (Celinder, 2005b). The online documentation standard is translated and the online documenta-
tion extended is in English only (Celinder, 2005b).

9.3 IFS’ document types

IFS online documentation consists of process models and the texts types that IFS calls window descriptions, activity descriptions, about descriptions and process descriptions. These texts often exist in two variants—one for the Windows platform and one for the web user interfaces (Celinder, 2005b). A brief explanation of the text types most commonly produced for IFS online documentation follows:

![Figure 9.2: IFS text types](image)

**Window Description** A description of a specific window, tab window or dialog box in IFS Applications. In the user documentation for IFS Applications there are 7,400 window descriptions which makes it the most common document type; see Figure 9.2.

**Activity Description** A description of how to perform a specific user action in IFS Applications. There are 4,400 activity descriptions in the documentation for IFS Applications.

**About Description** A general description of a business object in the business process. It may also be a description of a work flow or a specific function or functionality in IFS Applications. There are 700 activity descriptions in the documentation for IFS Applications.
Process Description A description of a process model with detailed information, which cannot be illustrated in a model. There are 500 process descriptions in the documentation for IFS Applications.

Web Page Description A description for the Web application, equivalent to the window description for the Windows application. It describes a specific page, tab page, or closely related page and gives the reader a very short and concise overview of area of use and available options in the page.

Portlet Description An explanation of what the portlet\(^2\) displays, its area of use, and available options.

9.4 IFS’ documentation process

People working with documentation at IFS today are scattered all over the company which makes it hard to make decisions about the language used in documentation. When documenters work in small groups, it is easier to make decisions about language rules that all the writers will follow; e.g., to use \textit{enable/disable} instead of \textit{check/uncheck} (Gabrielsson, 2005).

There exists a methodology at IFS for the creation of online documentation. There is also a description of roles in the development process concerning package development and system test. At IFS there exist a specific role in the project organization called the documenter. This role is performed by a technical writer or an application consultant. The documenter writes on-line documentation. In order to write documentation, the documenter needs to test the developed solution. The focus of the test is mainly on understanding the functional flows and to report the bugs found.

9.5 IFS Style and Writing Guide

According to IFS language strategy, all external communication requires consistency in the way writers at IFS express themselves (Ehlinger-Janzon, 2005). In order to make it easier to follow the language strategy, IFS has created the \textit{IFS STYLE AND WRITING GUIDE} as a reference for anybody writing or editing documents in English created within IFS (Ehlinger-Janzon, \footnote{A portlet is a re-usable Web component that displays relevant information to portal users; e.g., e-mail, news, or discussion fora.})
The main driving force for establishing these guidelines is that common language will strengthen the company’s global brand and promote the company’s position as an international company (Ehlinger-Janzon, 2005). The IFS Style and Writing Guide has two purposes (Ehlinger-Janzon, 2005). The first purpose is to make the publishing process from document creation to final review and releases as efficient as possible by addressing basic aspects of writing; i.e., grammar, usage, and punctuation (Ehlinger-Janzon, 2005). By providing these guidelines, IFS writers and reviewers all over the world are able to work from the same standard (Ehlinger-Janzon, 2005). The second purpose is that the guide makes it easier to produce marketing collateral and end-user documentation (Ehlinger-Janzon, 2005).

The style standards presented in IFS Style and Writing Guide are based on IFS-specific guidelines, the Associated Press Style Manual, and The Chicago Manual of Style (IFS, 2004c). The guide is maintained and updated by CMC Language board at IFS.

The chapters about writing technical material in the IFS Style and Writing Guide address guidelines and terminology associated with user documentation and other writing produced for IFS R&D. It includes R&D-specific grammar and usage rules. Also the guide includes explanations of how to refer to all the visual components within all types of IFS Application user interfaces. The purpose of R&D material is to provide instruction on how to install, configure, or use IFS Applications.

9.6 Known problems in IFS documentation

In order to gather as much information as possible to facilitate our case study, we have conducted interviews with four IFS employees currently working with writing, correction and translation of documentation at IFS; see Section 1.4.2 for more information about the persons interviewed. We asked the employees to describe the most common comprehensibility and transalatability problems that they encountered and also some possible solutions to those. Based on the interviews, a number of problematic issues with the documentation at IFS were summarized. We present them in the following sections.

---

3Marketing collateral is a term used for printed materials used to present information about a business and its capabilities, products or services. It can include brochures, fliers, fact sheets, direct mail pieces, and other communications pieces that are produced directly by the company (Women’s Business Center, 2001).

4Corporate Marketing & Communications
9.6.1 Sentence length and complexity

One of the main problems concerning comprehensibility and translatability in IFS documentation, according to Lovendhal (2005) and Holmberg (2005), is the sentence length and complexity. Lovendhal (2005) explains that the reason for sentences often being too long is that too many activities are explained in a single sentence. According to Holmberg (2005) the problem with long complex sentences is sometimes the result of unclear and not followed writing rules.

Lovendhal (2005) suggested that a way to deal with this problem is to give the writer a warning message if a sentence gets too long. Gabrielsson (2005) also suggested that an important way to help the writer during writing of documentation would be to think of a good sentence length delimitation and to warn the writer if the sentence gets too long. Unnecessary filler words are also a comprehensibility and translatability problem that is related to sentence length. (Lovendhal, 2005)

9.6.2 Terminology

The use of unexplained or undefined terminology is another common problem in IFS documentation (Lovendhal, 2005; Holmberg, 2005). Problems with comprehensibility and translatability arise when different terms are used to address the same element when instead one term should be used generally. A similar problem is when the same English word or expression is being used for many different things (Lovendhal, 2005; Holmberg, 2005).

Gabrielsson (2005) has a lot of experience with terminology problems at IFS. One major problem lies in the use of cross-domain terms. It is difficult to know if a term is the same across the different domains, since a writer in one domain is not normally familiar with the other domains. This is why there is sometimes a tendency to divide one term into several different terms for each domain. An example of this is the term planner which is divided into distribution planner and HR planner even if they refer to the same concept.

In order to aid the writer in his work with terminology problems, Lovendhal (2005) suggests to use highlighting of certain technical terms and troublesome words, as described in the IFS Style Guide, and to suggest the accepted terms.

9.6.3 Other comprehensibility and translatability issues

The problems mentioned in Section 9.6.1 and Section 9.6.2 are the most common problems concerning comprehensibility and readability (Lovendhal,
2005; Holmberg, 2005; Gabrielsson, 2005). There are however more issues that the people we interviewed have noticed during their work with IFS documentation. We briefly mention some of them in the following paragraphs.

**Double negations** One type of comprehensibility issue in IFS documentation is the use of double negations; i.e., the occurrence of expressions where negations occur twice in the same sentence (Gabrielsson, 2005).

**Addressing the reader** Another problematic issue effecting comprehensibility, is that the reader is not addressed consistently and according to IFS Style and Writing Guide. Also the use of first person *we* in documentation should be disallowed (Gabrielsson, 2005).

**Text formatting** Formatting of elements; e.g., typesetting (bold and italics), the use of paragraphs, or the use of lists, is a problem in IFS documentation, according to Lovendhal (2005). If not properly formatted the element can be misinterpreted.

**Reference to specific graphical user interface elements** Sometimes the writer refers to specific graphical user interface elements and writes out the word *field* or *column* in the documentation; e.g., “in this column you can write” (Gabrielsson, 2005). This can be confusing to the reader if the interface has changed and the column that the documentation refers to does no longer exists. Also the use of expressions specific to the graphical user interface; e.g., *right-click*, *operations menu* should be used according to the standard given in the IFS Style and Writing Guide.

**Abbreviations** At IFS there are no standardized rules for the use of abbreviations in the source language (Holmberg, 2005). This causes difficulties in defining the rules for abbreviations in the target language (Holmberg, 2005).

**Spelling errors** Occasionally spelling errors can lead to change in meaning of the words or expressions in the documentation which result in mistranslations (Holmberg, 2005).

**Untranslatable expressions** Sometime the source language contains expressions or abbreviations that should not be translated without making the translator attentive to these special expressions. This is typical for technical expressions and company or association names (Holmberg, 2005).
References  Writers sometimes copy words or expressions from other parts of the documentation instead of making references. This leads to more work with the maintenance of the texts and also increases the risk for inconsistent source text and/or target language translation (Holmberg, 2005).

Graphics  Figures or graphics that contain text that need to be translated prolongs the translation process since the translated text must be readjusted or aligned in the graphics which requires manual work and possible hyphenation which can lead to vagueness (Holmberg, 2005).

Adjustments to existing translations  Small adjustments in the existing material increase translation costs since translation and review for these must be done on the entire material again (Holmberg, 2005).
Chapter 10

IFS plans

During the writing of this thesis, IFS has been working on a project called Translation & User Support. Within this project, the development of a term database is a major part. The article database and the product-management tools are also very important. Figure 10.1 shows the major parts of the Translation & User Support project.

![Diagram](image)

Figure 10.1: Translation and User Support project at IFS

10.1 IFS term database

During spring 2005, IFS has been developing a term database as a part of the Translation & User Support project. The goal is to eliminate semantic ambi-
guities and thereby provide better translations (Hammarström, 2004b). The term database is designed to ensure that the same term is used consistently for the same concept in all documentation.

IFS defines a term as follows:

[A term is] one or several words with a precise meaning, in a specific domain. (Celinder, 2005d, p.12)

### 10.1.1 Design

![Figure 10.2: IFS term database](image)

Here is a list of the fields in IFS’ term database. After each description the data for the term *Customer Warranty Type Template Identity* is written within parentheses as an example. The layout is shown in Figure 10.2.

**term** This is the unique name of the term. It is never visible in the documentation. (*Customer Warranty Type Template Identity*)

**term domain** This is the domain where the term is unambiguously defined. Terms that are common for all IFS areas have their domain set to *general*. (*Distribution*)

**display names** These are the possible variants of the term that are used in the documentation.
full name This is how the term is written in its most explicit form. 
(Customer Warranty Type Template Identity)

abbreviation This is the shortest form of the term and might be an 
acronym. (Template ID)

short name This is a shorter form of the term, which can be used 
when there is not enough room for the full name. (Cust Warranty 
Type Template ID)

basic definition This is the description of the term. The term is often the 
name of a graphical component; this description is then the associated 
help text. (The code value that identifies the customer warranty type 
template.)

Term definitions

Since the focus in the IFS term database development is help texts, the 
descriptions are not written to define the term as precisely as possible, but 
to inform the user of how the component that is related to the term should 
be used.

According to Lovendhal (2005), a good term definition will usually consist 
of the following criteria:

• What is this?
• What does it look like?
• Where does it come from?
• Why is it here?
• How does it work?

As a minimum most term descriptions should answer the first two questions 
and if applicable the rest of the questions (Eklund, 2005).

Usages

Since IFS uses its term database for generating help texts for graphical views 
of different kinds, each term might need to be explained in more than one way. 
IFS has solved this by having multiple usages; each with its own “term basic 
definition”. These usages for instance correspond to whether the graphical 
element is a read-only field for displaying data, or if it is a field where the 
user should make a choice, or if the user is expected to enter new information 
in the field (Celinder, 2005a).
Other fields
Apart from the fields mentioned above, the term database is designed to support limited grammatical information such as plural forms and genitive forms. This shows that the database is designed for nouns. However, none of these fields are used today.

There is another field—"intention"—that is not currently used, which was added to enable the term creator to describe exactly what was meant in order to distinguish the term from similar terms (Hammarström, 2004a).

There are other fields as well that are related to version management and similar functions but we do not find them relevant to this discussion.

Translations
One objective of the term database is to make translation easier. It is possible to enter any number of translations to a description as needed. Since the term database is multilingual, translations of the terms will be done only once.

10.1.2 Extraction of terms
In order to fill the term database, IFS developed an import tool. This tool searched through the documentation that already existed as html-files and identified help texts associated with graphical fields—called field descriptions. The field names were then suggested as terms and the field descriptions as basic term definitions. This process was controlled by technical writers, who did choose the best existing field description as definition. Since the documentation until now had not been standardized, one field could have different descriptions in different help pages (Gabrielsson, 2005).

In the future, this import tool will not be used. When a new term is needed, it should be entered into the term database directly together with its necessary data, such as description. The term database will in this way always be up to date (Hammarström, 2004a)

10.1.3 Term relations
IFS has realized that it might be beneficial to be able to connect terms semantically to each other in the term database. Therefore, the term database supports relations such as "broader term", "related term", and "use for". These relations can be used to construct thesauri, but also to make sure that deprecated terms are substituted for the new replacement.
10.2 Controlled languages at IFS

When IFS started to work on the term database, it was seen mainly as a way to make documentation work more efficient. However, IFS realized the potential to continue with a controlled language. If such a controlled language would be developed, the term database should be used for the controlled vocabulary part (Celinder, 2005d).

In this case study we focus on the other part of a controlled language—the controlled grammar that should be used together with the vocabulary. Since this controlled language should be used when writing documentation and since IFS generates its documentation from the term descriptions, the text that will be controlled is actually these term descriptions in IFS’ term database.

To be able to verify that a text is written according to the controlled language, it must be possible to check that no disallowed words are used. This means that IFS needs to add common words to its term database or in another way construct a full vocabulary.

Since making a full vocabulary requires a lot of work, there might be ways of improving the documentation quality with respect to comprehensibility and translatability by using some but not all of the components of a controlled language. With a tool, the author can get constructive help when writing new term descriptions. For instance, term relations such as “use for” can generate a warning if the writer chooses a term that should not be used. Some of the controlled language grammar can also be tested easily by such a tool. This will not make it possible to say that a text is written in a controlled language, but it can possibly improve the documentation quality and be a step towards a full controlled language.

In this case study we attempt to construct a full set of controlled language rules. We also discuss how these rules can be integrated into a checking tool in the future.
Chapter 11

Rule suggestions

In this chapter we discuss the language rules that most of the existing controlled languages have in common. These are compared to recommendations from the style guides that we have studied. We then compare the rules to the needs of IFS, which results in rule suggestions. The rules are discussed from readability and translatability viewpoints.

In order to find rules to test, we have used a variety of sources. Here is a short summary of our sources. More detailed information is available in our bibliography.

Controlled languages We have studied the entire rule sets of AECMA Simplified English, Ericsson English, and PACE.

Style guides We have investigated recommended rules from the Chicago Manual of Style, Microsoft Manual of Style, Plain Language and IFS Style and Writing Guide.

Research reports We have used a number of research reports containing controlled language rules.

Technical writers We have interviewed experienced technical writers at IFS to find out what rules they believe to be useful, and also what they had to say about the rule suggestion from the other sources.

These sources have lead us to a large collection of rules. We have grouped the rules since many of them were very similar in how they are intended to control the language. Each section in this chapter corresponds to such a group.

For each group, we tried to choose one or a few rules that we thought could be interesting to test at IFS. In some cases, we modified the rules slightly to better match the other rules or to better fit the goals of IFS.
11.1 Approved words

A controlled language consists of a vocabulary and a set of rules, as mentioned in Section 5.1. One obvious such rule is to use that vocabulary to some extent. The exact formulation of the rule depends on the design of the vocabulary.

As mentioned in Section 7.5, the vocabulary of Ericsson English (EE) is divided into two categories—in some documents only words from one category are allowed (Ericsson, 1983). Unlisted words can be included if they are explained by EE words.

In AECMA Simplified English (SE), technical names and technical verbs can be used in addition to the words listed in the vocabulary. Some examples of technical names are names of tools and materials, scientific terms, colors, and numbers. Technical verbs are typical actions performed by the reader. In computer programs, this can be different verbs associated with the graphical user interface; e.g., clear, close, drag, highlight, and scroll (AECMA, 2004).

IFS is developing a term database and has the following goal (Hammarström, 2004a)

Objectives: [...] A consistent and translatable language based on a well defined term catalogue [...] 

We therefore suggest the following rule:

**Rule 1** Only terms that are approved in the IFS Term Database are allowed.

11.2 Sentence length

O'Brien (2003) writes that only one type of rule was common to all of the controlled languages that she examined—sentence length limitation. How long a sentence should be to be maximally readable is a standard readability question, according to (Platzack, 1989). The answer is neither too long nor too short (Platzack, 1989). In order to get a more specific answer we take a look at what existing controlled languages recommend.

Plain language recommends the use of short and medium-length sentences. As a guideline, it is recommended to keep the average sentence length to about 20 words (PLAIN, 2004). In AECMA, the maximum length of a procedural sentence is 20 words, while descriptive writing may contain sentences of up to 25 words (AECMA, 2004). In Ericsson English (EE) as well as in Maxit’s Controlled English, short sentences are desired (Cremers, 2003; Ericsson, 1983). One of the rules in PACE is that sentences should be kept short (Nyberg et al., 2003).
Møller (2003) writes that shorter sentences are preferred by machine translation systems. However, there is an important exception—if splitting up sentences involves introducing pronouns, there is a risk of introducing difficulties related to anaphora resolution. In those cases, the original sentence might be easier to handle for the machine translation system.

AECMA (2004), on the other hand, stresses the importance of varying sentence length to make the text interesting to read.

IFS Style and Writing Guide recommends that the writer does not use more than 17 words as average length for a sentence (IFS, 2004c). This guide also stresses the importance of variation in the length of the sentences in order to make the writing more interesting (IFS, 2004c).

We suggest the following rules:

Rule 2 The average sentence length should be 13–17 words (if the text consists of at least four sentences).

Rule 3 All sentences should contain no more than 20 words.

Texts consisting of three sentences or less have no average sentence length limit, since the need for variation in such a short text probably does not compare to the effort of rephrasing it.

11.3 Sentence complexity

Complex sentences are tiresome for most readers in that they reduce reading speed and comprehensibility (Melin, 2004). One type of complicated sentence structure that makes reading more difficult is nesting of subordinate clauses in sentences (Gunnarsson, 1989).

In Ericsson English (EE), sentences should have few subordinate clauses (Ericsson, 1983). A maximum of one adverbial clause in a sentence is recommended in EE, and these clauses should never be used in the middle of a sentence (Ericsson, 1983). If a sentence in EE has more than one comma, it is probably too long, unless the commas are used to separate items in a list (Ericsson, 1983).

In Maxit’s Controlled English, there is a rather vague rule saying that sentences should be simple (Cremers, 2003).

EE uses that to introduce relative clauses, as in Example 11.1. EE does not use which in the same way; sentences requiring which should be split in two (Ericsson, 1983).

Example 11.1 Press the button that is located in the bottom left corner.
AECMA Simplified English (SE) suggests using vertical lists instead of listing several items or instructions in one sentence (AECMA, 2004).

MSTP recommends avoidance of long and complex sentences (Microsoft Corporation, 1998). The reason for this according to MSTP, is that a complex sentence requires the translator to first understand the content; then to choose the main clause, which the translation should begin with; and then translate it (Microsoft Corporation, 1998). When conveying ideas, MSTP recommends to put the main ideas first and to break the material into small units (Microsoft Corporation, 1998).

In order to limit the complexity of sentences, we suggest the following rule:

**Rule 4** A maximum of one subordinated clause is allowed in a sentence.

### 11.4 Paragraph length

In AECMA Simplified English (SE), the maximum length of a paragraph is six sentences (AECMA, 2004). Each paragraph should have only one topic, and the first sentence should help the reader to understand what this topic is (AECMA, 2004). SE also has a rule disallowing one-sentence paragraphs more often than once per ten paragraphs (AECMA, 2004).

IFS Style and Writing Guide states that paragraphs should typically be short and concise, and that the length of paragraphs may vary, but generally 100 to 150 words is a good average (IFS, 2004c).

Since limitations on paragraphs seem appropriate, we suggest the following rule:

**Rule 5** All paragraphs should contain no more than 6 sentences.

Each item of a vertical list counts as a paragraph itself. There is no limit of the allowed length of such a list.

### 11.5 Sequences

In Ericsson English (EE), sequences of events or activities should be described in the order in which they occur (Ericsson, 1983). The same rule applies in PACE (Nyberg et al., 2003).

According to Kirkman (1992), it is clumsy to group several instructions in one long paragraph. Instead, it is much clearer to write instructions as
Chapter 11. Rule Suggestions

A sequence of short and imperative instructions that describe the order of events that actually occur in an activity.

Example 11.2 is dangerous, since the user might read the first words and restart the computer before realizing that the work should have been saved. Example 11.3 avoids this problem.

Example 11.2 Restart the computer after you have saved your work.

Example 11.3 After you have saved your work, restart the computer.

Rule 6 Always write events in the order that they are supposed to occur.

According to this rule, Example 11.2 is not allowed.

11.6 Contractions

A contraction is a word formed by two or more words by omitting or combining sounds. In English, these omissions are denoted by the use of an apostrophe; e.g., it's is a contraction of it is.

Plain language recommends writers to use contractions when appropriate (PLAIN, 2004). The reason for this is that contractions help the reader by increasing reading speed and soften the tone in the documents produced. In EE, however, the use of contractions is disallowed, partly because of inaccuracy (Ericsson, 1983).

Microsoft Corporation (1998) writes that the use of contractions is at the discretion of the editor. Some advantages of contractions are that they lend an informal tone and save space. Concerning disadvantages, Microsoft Corporation (1998) writes that readability experts claim that it takes more time to decode a contraction than a complete phrase during reading. CMS does not give advice about the use of contractions, but only states that contractions are more natural.

IFS Style and Writing Guide states that all forms of contractions should be avoided in technical writing (IFS, 2004c). The guide explicitly recommends that the writer do not use contractions of helper verbs and the word not; e.g., can't and isn't (IFS, 2004c). Also, the use of contractions of personal pronouns and verbs should be avoided; e.g., you're (IFS, 2004c).

Based on the current IFS recommendations, and the support of EE concerning clarity and MSTP concerning reading time, we suggest the following rule:

Rule 7 Contractions may not be used.
11.7 Compound terms

When several words together form a term we have what is called a compound term. This is also called word clusters or—in the common case where only nouns are put together—noun clusters. Compound words exert our short-term memory and make reading and understanding more difficult (Melin, 2004).

Møller (2003) found that in machine translation, compound terms are preferred over rewritten phrases; e.g., Example 11.4 will be more accurately translated than Example 11.5.

Example 11.4 a straw retarding curtain

Example 11.5 a curtain that retards the straw

AECMA (2004) generally disallows the use of compounds consisting of more than three nouns. However, if a longer compound term is part of the company’s official terminology, it is accepted, but it should be clarified with hyphens or an explanation; see Example 11.6 and Example 11.7.


Example 11.7 Write: Engage the rump service door safety connector pin (the pin that holds the rump service door, referred to in this procedure as the safety connector pin) before you do this procedure. (AECMA, 2004, p.1-2-3)

In Maxit’s Controlled English, one of the rules says that word clusters should be avoided (Cremers, 2003).

O’Brien (2003) found that a majority of the controlled languages that she examined forbids the use of noun clusters longer than three nouns.

IFS Style and Writing Guide does not give any recommendations about the length of compound terms versus rewritten phrases. The guide does however recommend the use of hyphens when forming noun clusters (IFS, 2004c).

The main reason for limiting compound length seems to be the ambiguity problems of which word determines which. We suggest the following rule:

Rule 8 Compound terms may not consist of more than three words.

Exception A: The rule does not apply if the entire compound is an approved term in the IFS Term Database.

Exception B: The rule does not apply if the compound consist of words that are visually grouped together into units and there are no more than three such units. Each unit must itself meet the requirements of a compound.
Example 11.8 This tab explains the Create Company Component Log window.

Example 11.8 shows a sentence with a compound of length four. However, it contains two units of words that are visually grouped together by case (the first words are capitalized initially):

1. Create Company Component Log
2. window

The first unit has four words, but is valid it is an approved term in the IFS Term Database according to Exception A. The second unit is a single word and therefore valid. Hence, the sentence in Example 11.8 is valid because of Exception B.

11.8 Articles

AECMA (2004) states that articles should not be omitted before nouns. This applies to:

- definite articles (a, an)
- the definite article (the)
- demonstrative adjectives (this, these).

Møller (2003) agrees that definite articles should not be omitted, since the articles help in machine translation. These articles also improve comprehension for both experts and non-expert readers (Møller, 2003).

CTE also strongly recommends the use of articles before nouns for improving comprehensibility and translatability (Nyberg et al., 2003).

O’Brien (2003) found that a majority of the controlled language rules that she examined require the use of articles before nouns.

As a general rule, IFS Style and Writing Guide recommends the use of articles; e.g., a, an, and the before nouns in full sentences (IFS, 2004c).

Since all sources seem to agree, we suggest:

Rule 9 Articles (a, an, the, this, these) may never be omitted.
11.9 Verb forms

A verb can be said to have five properties: tense, mood, voice, person, and number (University of Chicago Press, 2003). Number is only singular or plural, and not very interesting to control. The other properties, on the other hand, are discussed in this section.

11.9.1 Tenses

In English, verb tenses can be divided into three time frames and four aspects, which makes twelve tenses, as shown in Table 11.1.

<table>
<thead>
<tr>
<th>Verb form</th>
<th>past</th>
<th>present</th>
<th>future</th>
</tr>
</thead>
<tbody>
<tr>
<td>simple</td>
<td>went</td>
<td>go</td>
<td>will go</td>
</tr>
<tr>
<td>progressive</td>
<td>was going</td>
<td>is going</td>
<td>will be going</td>
</tr>
<tr>
<td>perfect</td>
<td>had gone</td>
<td>has gone</td>
<td>will have gone</td>
</tr>
<tr>
<td>perfect progressive</td>
<td>had been going</td>
<td>has been going</td>
<td>will have been going</td>
</tr>
</tbody>
</table>

Ericsson English (EE) only allows the three simple tenses (Ericsson, 1983). EE also states that the present tense should always be used when possible (Ericsson, 1983). AECMA Simplified English (SE) allows the same tenses.

IFS Style and Writing Guide recommends to always use the simple present tense unless there is a need to draw a distinction between the timing of two or more events (IFS, 2004c). The guide states that neither future-tense verbs nor present progressive tense is appropriate (IFS, 2004c). According to IFS Style and Writing Guide, the assumption is that IFS material describes what typically happens as it happens, which is accomplished by present tense (IFS, 2004c). IFS Style and Writing Guide also states that present tense makes the text more readable because it eliminates the need to use auxiliary verbs (IFS, 2004c).

MSTP recommends the writer to use the simple present tense as much as possible and avoid all other tenses unless there is a need to describe events in terms of the past or future (Microsoft Corporation, 1998).

There seem to be a general agreement considering tense, why we suggest:

Rule 10 Use only the simple verb tenses.

Rule 11 Use simple present tense unless there is a need to distinguish between the timing of events.
11.9.2 Mood

The mood of a verb describes the attitude of the speaker toward the subject; see Section 4.5.5 for a discussion of moods in English.

AECMA Simplified English (SE) uses indicative and imperative mood (AECMA, 2004). Although not explicitly stated, these forms are used in Ericsson English (EE) as well.

According to Microsoft Corporation (1998), the indicative mood should be used to express general information such as facts and explanations. The imperative mood should be limited to procedures and direct instructions (Microsoft Corporation, 1998). Since the subjunctive mood often expresses wishes, hypothesis, and conditions contrary to facts, MSTP recommends that this mood should be avoided as much as possible.

We suggest the following rules:

Rule 12 Instructions should be written using the imperative mood

Rule 13 Descriptions should be written using the indicative mood.

11.9.3 Voice

Voice shows whether the subject acts (active voice) or is acted on (passive voice); see Section 11.9.3.

Ericsson English (EE) and Maxit’s Controlled English use only the active voice (Cremers, 2003; Ericsson, 1983), and O’Brien (2003) found that a majority of the controlled languages that she examined enforces the use of active voice.

In AECMA Simplified English (SE), the active voice must be used in procedural writing, and should be used as much as possible in descriptive writing (AECMA, 2004). Plain language recommends that the majority of sentences in a document be in the active voice in order to avoid ambiguity in responsibilities and to clearly express who is responsible for what (PLAIN, 2004).

MSTP recommends that the writer use the active voice in general and that the passive voice should be avoided except when necessary to avoid a wordy or awkward construction (Microsoft Corporation, 1998). The CMS states that as a matter of style, passive voice is typically, though not always, inferior to active voice (University of Chicago Press, 2003).

Møller (2003) showed that in general, machine translation systems have no difficulties in translating passives. She also writes that experts often prefer the passive voice. However, most non-experts favor active voice.
IFS Style and Writing Guide states that it is better to use active rather than passive voice (IFS, 2004c). The reason for this, according to IFS Style and Writing Guide, is that active voice is more precise and requires fewer words (IFS, 2004c). Passive voice should only be used if the agent is irrelevant or unknown (IFS, 2004c).

Our opinion is to disallow passive voice, but we do not need an explicit rule for this, since it is governed by the rule concerning past participles; see Section 11.10.2.

11.9.4 Person

Although not explicitly stated in Ericsson (1983) or AECMA (2004), the examples in these documents suggest that the reader is addressed with you; i.e., the second person form should be used as much as possible. This concerns descriptive texts, since instructions are written in imperative form and therefore do not contain a subject.

According to Microsoft Corporation (1998), the second person should be used in general when referring to the user, since it focuses the discussion on the user and makes it easier to avoid the passive voice (Microsoft Corporation, 1998).

MSTP recommends the avoidance of the use of first person (I and we), because it can sound patronizing (Microsoft Corporation, 1998).

IFS Style and Writing Guide states that the best way to engage the user is to use the second person, as much as possible (IFS, 2004c). The third person should be avoided and only be used when the writer is writing to another person than the user of the computer or program, such as a consultant or developer (IFS, 2004c).

We suggest the following rule:

**Rule 14** Descriptions should be written in second person form when there is a need to refer to the user.

11.10 Verbals

Verbals are verb forms that act as other parts-of-speech. A common example of this is the participles, that behave as adjectives.

**Example 11.9** adjective: Look at the red text.

**Example 11.10** present participle: Look at the blinking text.
Example 11.11 past participle: *Look at the highlighted text.*

Participles as well as gerunds and infinitives are discussed in this section.

### 11.10.1 Present participle

A present participle is the *ing*-form of a verb, when functioning as an adjective. The present participle is the head word of the present participle phrase, which can be removed from the sentence without making it ungrammatical.

In the PACE language, *ing*-forms should be avoided completely (Nybørg et al., 2003). Neither AECMA (2004) nor Ericsson (1983) allow the present participles; therefore Example 11.12 must be re-written to something similar to Example 11.13 or Example 11.14.

Example 11.12 The menu, *fading quickly*, re-appears if you click on the button.

Example 11.13 The menu, which fades quickly, re-appears if you click on the button.

Example 11.14 The menu fades quickly but re-appears if you click on the button.

In her study, Møller (2003) showed that the re-written forms without present participles are more easily handled by machine translation systems. Since present participles seem to be problematic, we suggest the following rule:

**Rule 15** Present participles are not allowed.

### 11.10.2 Past participles

A past participle is the past tense of a verb when functioning as an adjective.

Ericsson English (EE) allows constructions with past participles, but the use of past participles in EE is limited to describing a state (Ericsson, 1983).

Example 11.15 *If the window is closed, you can open it again.*

Example 11.16 The window is *closed* by pressing the exit button.
In Example 11.15, the word "closed" is used as an adjective, and is accepted in EE as long as it is a way of saying that someone in the past has closed the window (Ericsson, 1983). Example 11.16 shows how the same form could be used as describing the event of closing the window; this usage is not allowed in EE since it is a passive construction.

AECMA Simplified English (SE) has the same restriction on the use of past participles as EE (AECMA, 2004). It is clarified that these past participles must be preceded by a form of the verb to be or become (AECMA, 2004).

Based on the agreement concerning the use of past participles, we suggest the following rule:

**Rule 16** Past participles are allowed only when describing a state and either precedes the noun it determines or follows a form of to be or become.

An instruction as Example 11.17 is thus also valid.

Example 11.17 *Click on the displayed button.*

### 11.10.3 Gerunds

The gerund is the *ing*-form of a verb, when functioning as a noun, as in Example 11.18.

Example 11.18 *Customizing the application is important.*

Maxit’s Controlled Language disallows the gerund form (Cremers, 2003). O’Brien (2003) writes that a majority of the controlled languages that she examined disallow the gerund form. To avoid the gerund, Example 11.18 has the to be rewritten to something similar to Example 11.19.

Example 11.19 *It is important to customize the application.*

Neither AECMA Simplified English (SE) nor Ericsson English (EE) allows the constructions of gerunds. However, both of these languages do permit certain gerunds that are technical terms (AECMA, 2004; Ericsson, 1983). For instance, EE allows words as *testing* and *lacing*.

MSTP does not disallow the use of gerund but only warns the writer that the use of gerunds in text can be ambiguous, which causes problems for the translator (Microsoft Corporation, 1998).

Since the gerund form is often disallowed, we suggest the following rule:

**Rule 17** Gerunds are not allowed unless explicitly listed as a term in IFS Term Database.

Since we have already forbid present participles, this means that no *ing*-forms are allowed if not in the IFS Term Database.
11.10.4  **Infinitives**

An infinitive is a verbal consisting of the word *to* plus a verb (in its stem form) and functioning as a noun, adjective, or adverb; see Example 11.20.

**Example 11.20**  *The dialog asks you to complete the form.*

Infinitives are not restricted in any sense in AECMA Simplified English (SE). Ericsson (1983) does not mention infinitives, but from the examples it is clear that infinitives are a part of Ericsson English (EE).

We see no reason to limit the use of infinitives.

11.11  **Phrasal verbs**

In English, there are many word combinations of a verb and a particle, which is often a preposition or an adverb. These **phrasal verbs** are sometimes called **two-word verbs**. It is easy to distinguished them in speech, since the particle is emphasized; compare Example 11.21 to Example 11.22.

**Example 11.21**  *turn on the radio*

**Example 11.22**  *turn on the spot*

In writing, phrasal verbs are a source of ambiguity, and CTE recommends the writer to use constructions without them (Nyb erg et al., 2003).

No other source that we have found discusses phrasal verbs. However, we find the observation interesting and suggest the following rule:

**Rule 18**  *Phrasal verbs are not allowed.*

11.12  **Ellipsis**

An important property of a text that influence comprehension is cohesion; see Section 2.3.3. The unit of analysis for cohesion is the cohesive tie. **Ellipsis** is such a tie; it is the omission of a part of a clause because the part is implied by the context\(^1\). The result is a clause that is still understandable, but not strictly grammatical; see Example 11.23.

**Example 11.23**  *The label is connected to the button and the button [is connected] to the form.*

---

\(^1\)Ellipsis is also the name of the symbol “...” but this sense is not used here.
According to AECMA (2004), words should not be omitted to make sentences shorter. CTE writes that the use of ellipsis should be ruled out whenever possible, since it introduces potential ambiguity in ellipsis resolution (Nyberg et al., 2003). PACE explicitly disallows elliptical constructions (Nyberg et al., 2003).

In her study, O'Brien (2003) found that a majority of the controlled languages at least forbid the omission of relative pronouns; see Example 11.24.

Example 11.24 This is the text [that] you selected.

Møller (2003) showed that the occurrence of ellipsis makes texts less comprehensible for both experts and non-expert readers. Machine translation systems also have trouble handling ellipsis.

MSTP does not explicitly give advice about ellipsis but recommends the writer to be sure that all necessary words are included to make the meaning clear (Microsoft Corporation, 1998).

Since our sources agree, we suggest the following rule:

Rule 19 Elliptical constructions are not allowed.

11.13 Negatives

In Ericsson English (EE), unnecessary negatives should be avoided (Ericsson, 1983). For instance, EE recommends that Example 11.25 should be rewritten as Example 11.26 (Ericsson, 1983, "Instructions" p.13).

Example 11.25 Do not remove the device until the machine has stopped.

Example 11.26 Stop the machine. Then remove the device.

In Maxit's Controlled English, statements should be written in positive form (Cremers, 2003).

MSTP recommends the avoidance of negative constructions; they recommend that information be stated positively whenever possible (Microsoft Corporation, 1998). The documentation should tell the user what to do, not what not to do (Microsoft Corporation, 1998). According to MSTP, the use of not can make a sentence harder to read and understand (Microsoft Corporation, 1998).

We believe that negations should be allowed, but used carefully. To avoid obscure sentences, we suggest the following rule:

Rule 20 A sentence may not contain more than one negation; e.g., no, not, nothing, neither, never.
Chapter 12

Rule analysis

In this chapter, we discuss how well our rules have worked in practice. We analyze whether it is simple to use the rules without any problems; we call this rule applicability. Then, we analyze the effects of the rules in order to see if the readability and translatability has increased because of our suggested rules. In order to study the applicability and effect of the rules, we started with selecting test documents. We begin this chapter by discussing this selection.

12.1 Choice of test documents

In order to test our rules, we selected a number of documents from the IFS documentation. These were all versions that had not yet been sent for review. To get a representative text sample for the test of controlled language rules, we chose to put together a mixture of the most common text types in IFS documentation: About Descriptions, Window Descriptions, and Activity Descriptions.

Apart from the text types, we also mixed texts based on origin; we used texts from different domains of IFS Applications; e.g., Financials and Distribution, texts written by different writers in different countries; e.g., Sweden and Sri Lanka, and texts of different length. After we had decided the criteria for this mixture, we selected texts at random from the candidates.

We have included our text samples in Appendix D.

12.2 Rule applicability

In this section we test the applicability of our rules. By this we mean that we investigate if there are any difficulties in the analysis and the rewriting
of the text. During the text analysis, it is necessary that it is possible to tell which rules are followed and which are broken. Then it must be possible to rewrite the text so that all rules are followed without changing the message of the text.

12.2.1 Text analysis

In order to examine if the original IFS texts obeyed our suggested rules, we analyzed the text manually. To make sure to minimize our mistakes, we made one analysis each before we compared, discussed and agreed on an analysis that we both supported. During these discussions, we found a few cases where one of us had not realized that a rule had been broken. However, we always agreed on the interpretation of the rules, which gives some indication of their clearness.

Since more than one rule can modify the same aspects of the texts, there is a possibility that there are two rules that, when applied separately, both give positive results, but when applied together makes the texts worse. There is also a possibility that there are no ways to combine the rules and still be able to communicate the intention of the text. For this reason, we applied all rules simultaneously.

During the analysis of texts based on counting words, calculating average sentence length and paragraph length it was practical and simple to get help from tools in Microsoft Word; e.g., word count and grammar check. However, none of the rules were difficult enough to require such tools.

We here give a short comment to each rule based on our text analyses. For a complete list of our suggested rules, see Appendix A. When we write that a rule is simple or difficult to apply, we mean that it is simple or difficult to see if that rule is obeyed or broken in a piece of text.

R1 Since IFS is still developing its term database, we could not use it to test that only approved terms were used in the text. Therefore, we did not use this rule in our analysis. If IFS expands its term database to a controlled vocabulary, this rule could easily be used and tested.

R2 Average sentence length is easy to measure with a tool such as the grammar checker in Microsoft Word. This rule requires that it is possible to tell what makes a sentence a sentence. When a list of sentences is introduced by a piece of text ending with a colon, we have treated that piece as a sentence. If a list of items is introduced by a sentence and a colon, we have treated the introductory text and the list of items as one sentence. AECMA (2004) devotes an entire section to word count
in tricky situations that we recommend for this purpose. The rule was broken in several cases.

R3 This rule is easy to apply and one of the most commonly broken ones. See the discussion above (on R2) concerning what is a sentence.

R4 This rule is quite easy to apply, but it requires knowledge of subordinated clauses. It was not broken many times. We only found one occurrence where this rule was broken but not the rule above (R3). This indicates that the rule might not be very useful. The opposite—sentences that breaks the rule above (R3) but not this rule—was relatively common.

R5 This rule is easy to apply. See the discussion above (on R2) concerning what is a sentence. We treat a blank line as separating paragraphs. We also treat the items in a list as separate paragraphs unless they are only a part of a sentence. This rule was broken quite often.

R6 This rule was hardest to apply. When a number of actions should be taken in a given order it is not a problem. But it is hard to tell if a condition should precede its effect when the effect is desired and requires the condition.

R7 We never found any contractions, so this rule was never broken. However, it is an easy rule to apply since contractions are easy to detect.

R8 This rule requires two exceptions, so it looks quite complicated. However, it was never a problem to use the rule other than that we could not use the IFS term database. We therefore had to reason about what we could expect to be a term and what was terms compounded with other terms or words. We believe that this rule will not be a problem when there is a term database to verify against. See further the discussion on R1 above.

R9 This rule was quite hard to apply as it was formulated. If an article omission makes the text ungrammatical, then it is not our responsibility to fix it, since we require correct English before we apply our rules. If an article omission does not make the text ungrammatical, then it is hard to say if it really is an omission. We did not find any article omissions in the texts that we analyzed.

R10 This rule was broken quite often, but it was easily applied. It requires knowledge of verb tenses.
R11 This rule is a bit vague since it requires that it is possible to see when there is a need to distinguish between the timing of events. This was not a big problem in the text analysis. The rule was broken quite often and there were only a few cases in the texts that required this distinction in timing.

R12 This rule requires that the distinction between instructions and descriptions is clear. This was not a problem in the text analysis, so it was quite easy to apply. The rule was broken a few times.

R13 This rule is the opposite of R12 above, and the same comments apply.

R14 This rule was never broken in any of texts that we analyzed. A reason for this is probably that IFS already has a policy of referring to the user as you. However, we believe that this is an easy rule to apply.

R15 This is a rule that is broken quite a few times. It is an easy rule to apply, since most verbs ending with ing are present participles. The exception are gerunds, but since they are only allowed if listed in the term database this possible confusion would not lead to any problems.

R16 This rule is the most commonly broken rule. Since it rules out most uses of past participles, text that is written in passive voice will break this rule continuously. It is easy to detect past participles, but it can require some thought to realize if it is describing a state or a process.

R17 This rule is easily applied with the aid of IFS term database. Since that database is not yet ready, we have considered all gerunds as breaking this rule. This has happened a few times during the text analyses.

R18 This rule was only applied once. However, it is an easy rule to apply; it only requires understanding of the text to tell if a construction of a verb and a particle is a phrasal verb or not.

R19 After further studies of ellipsis in order to find all existing variants, we realized that ellipsis is a very complex phenomenon. It is sometimes very difficult to say if a certain construction is elliptical or not. We did not classify any constructions as elliptical during the text analyses.

R20 This is naturally a very simple rule to apply. However, we found no occurrences of two negations in the same sentence.
12.2.2 Text rewriting

In this section we discuss the rewriting of the text that was necessary to make the text conform to all the rules that we have suggested. More explicitly we rewrote the text with regard to the rules that we had broken. We had to pay attention to not breaking any of the other rules in the process.

The rewriting was made individually and then we discussed our suggestions and agreed on a version that we both supported. Sometimes we had rewritten the problematic text in the exact same way, but in a majority of the cases we had different solutions to the problems. However, it was not hard to agree on a common solution.

We rewrote the text manually without any help from tools or algorithms. However, we used the grammar check in Microsoft Word for word counting purposes to make sure that we did not construct too long sentences.

Here is a list of comments on the rewriting process for each rule:

R1 We never rewrote the text according to this rule, but it would involve switching to approved terms or suggesting new terms to the IFS term database.

R2 This is a quite difficult rule to follow since it affects the entire text. However, when we rewrote the text according to the other rules, this rule was always obeyed in the end.

R3 This rule means shortening of sentences. Sometimes it was natural to split a sentence into its clauses. Sometimes it was trickier to get under the limit of 20 words and still preserve the flow of the text.

R4 It is quite easy to split a sentence with many clauses, so this rule was never hard to adjust to.

R5 The question of where to split long paragraphs arose because of this rule, but it never meant much trouble.

R6 When it was clear in which order the items were supposed to come, the rewriting was easy since it only involved a simple reordering.

R7 To spell out a contraction can probably be seen as the easiest way to follow a rule, even if this was never even necessary.

R8 If a compound violates this rule, we believe that it is easy to fix. We suggest splitting noun clusters or adding text formatting that helps in grouping the words.
R9 If an article is considered missing, it is trivial to insert it.

R10 This rule required the most difficult rewritings. Since perfect present tense is not allowed, it becomes very difficult to write conditions based on what could have happened in the past. Sometimes the simple past worked, but sometimes serious rewriting was needed.

R11 It is easy to rewrite text according to this rule. If simple present tense is preferred, then it is easy to change to it.

R12 This rule only requires the rewriting to the imperative mood, which is easy.

R13 This rule can require rewriting from imperative to indicative mood, which requires the addition of a subject. This is normally evident from the context, which makes this a fairly easy rule.

R14 This rule only requires simple rewriting, since it is easy to change a sentence from third to second person form.

R15 Rewriting a sentence to get rid of present participles requires some work, but is usually done without any problems.

R16 Getting rid of past participles can be a challenge, since it often requires the addition of an agent to the sentence. This can be difficult, since passive voice is often used when the author tries to hide the agent. Sometimes the agent is hidden because it is not important and sometimes it is hidden because the author assumes that the reader can figure it out. However, it is common that the reader does not realize who the intended agent is. We had trouble identifying some of the agents when we rewrote passive constructions to get rid of past participles in our sample texts.

R17 When the text contains a gerund that is not in the term database, one solution is of course to add it to the term database. This should only be done if there is no good alternative way to write the sentence. If there is a good alternative way, then the rewriting is simple. This rule did therefore not cause a problem to us.

R18 Phrasal verbs often have synonyms that are not phrasal verbs. In that case it is simple to pick such a synonym instead. In other cases it is possible to substitute it for a short explanation instead.
R19 If an ellipsis is identified, then it is easy to make it a reference instead or add the words that were left out.

R20 If a sentence contains double negations, it is probably an indication that the message can be expressed in a simpler way. It should not be hard to rewrite the sentence if it is clear what it tries to say. A simple solution is otherwise to split the sentence, so that each sentence contains a maximum of one negation.

We believe that future authors will have less problems than we had since some of the rules probably will not be broken. An author that is trained with our suggested rules will not attempt to write instructions without using the imperative mood, for instance. Some of the rules are not as easy to follow, such as the word limit, since it is not natural to think in number of words while writing a sentence. However, with practise the authors will probably not write sentences much longer than the maximum limit. And when they do exceed such limits, they have the possibility to rewrite the text while they are still focused on the message. We believe that this is important to consider and that it will make the rewriting process significantly easier.

12.3 Rule effects

In this chapter, we investigate the effects of our rules on the text. In other words, we study the difference between the source texts and the rewritten texts with regard to readability and translatability.

12.3.1 Theoretical analysis

A numerical text analysis of the different types of IFS texts is presented in Table 12.1–12.3. We present the analysis of the original texts together with the analysis of the rewritten texts.

As can be seen, after the rewriting the texts contained more but shorter sentences. The Flesh Reading Ease score became better and the grade level decreased. These two measures give a better rating on all the texts after applying our suggested rules.

It is also interesting to note that although the number of paragraphs and sentences have increased, the number of words are about the same in both versions. This is important since translation companies often charge per word.
### Table 12.1: Text analysis of About Vertex Integration

<table>
<thead>
<tr>
<th>Text version</th>
<th>Original</th>
<th>Rewritten</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of paragraphs</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>Number of sentences</td>
<td>22</td>
<td>29</td>
</tr>
<tr>
<td>Number of words</td>
<td>408</td>
<td>399</td>
</tr>
<tr>
<td>Longest paragraph (sentences)</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>Longest sentence (words)</td>
<td>39</td>
<td>20</td>
</tr>
<tr>
<td>Average of words per sentence</td>
<td>18.4</td>
<td>13.8</td>
</tr>
<tr>
<td>Flesh Reading Ease</td>
<td>40.5</td>
<td>42.6</td>
</tr>
<tr>
<td>Flesh-Kincaid Grade Level</td>
<td>11.9</td>
<td>10.7</td>
</tr>
<tr>
<td>Information density</td>
<td>2.91</td>
<td>2.39</td>
</tr>
</tbody>
</table>

### Table 12.2: Text analysis of Create Company In All Components

<table>
<thead>
<tr>
<th>Text version</th>
<th>original</th>
<th>rewritten</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of paragraphs</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Number of sentences</td>
<td>25</td>
<td>29</td>
</tr>
<tr>
<td>Number of words</td>
<td>383</td>
<td>360</td>
</tr>
<tr>
<td>Longest paragraph (sentences)</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Longest sentence (words)</td>
<td>44</td>
<td>20</td>
</tr>
<tr>
<td>Average of words per sentence</td>
<td>17.6</td>
<td>13.5</td>
</tr>
<tr>
<td>Flesh Reading Ease</td>
<td>43.2</td>
<td>48.8</td>
</tr>
<tr>
<td>Flesh-Kincaid Grade Level</td>
<td>11.3</td>
<td>9.6</td>
</tr>
</tbody>
</table>

### 12.3.2 Interviews

In order to examine in practise if the texts rewritten with our suggested controlled language rules improved the comprehensibility and translatability compared to the original IFS texts, we conducted two interviews with two documentation experts at IFS.

#### Preparation

After we rewrote the three texts, we prepared for two interviews. These interviews were intended to give us feedback on our suggested controlled language rules.

We divided each text into logical parts which could easily be found in both the original version and the rewritten version. We then put together two versions of each of the three texts. The first version consisted of the parts in order, but taken at random from the original and the rewritten version.
Table 12.3: Text analysis of Automatic Sourcing Candidates

<table>
<thead>
<tr>
<th>Text version</th>
<th>original</th>
<th>rewritten</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of paragraphs</td>
<td>17</td>
<td>22</td>
</tr>
<tr>
<td>Number of sentences</td>
<td>35</td>
<td>46</td>
</tr>
<tr>
<td>Number of words</td>
<td>784</td>
<td>803</td>
</tr>
<tr>
<td>Longest paragraph (sentences)</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>Longest sentence (words)</td>
<td>31</td>
<td>20</td>
</tr>
<tr>
<td>Average of words per sentence</td>
<td>17.5</td>
<td>13.6</td>
</tr>
<tr>
<td>Flesh Reading Ease</td>
<td>43.9</td>
<td>48.0</td>
</tr>
<tr>
<td>Flesh-Kincaid Grade Level</td>
<td>11.3</td>
<td>9.9</td>
</tr>
</tbody>
</table>

The second version consisted of the remaining parts. The purpose was that it should not be evident which parts were original and which parts we had rewritten.

The two versions of each texts are included in Appendix E. Compare these texts to the originals and the rewritten versions in Appendix D to get a better understanding of our procedure.

Realization

A total of 14 parts were evaluated. For each part, the two interviewed persons were asked to choose the version that they preferred. We stated that the goal was to make the text as comprehensible and translatable as possible. Since none of the persons were experts in these areas, many decisions were made based on a gut feeling. However, both persons are experienced technical writers, which makes their opinions valuable.

Result

Table 12.4: Result of interviews

<table>
<thead>
<tr>
<th>preferred:</th>
<th>Person A</th>
<th>Person B</th>
<th>A+B</th>
</tr>
</thead>
<tbody>
<tr>
<td>originals</td>
<td>1 (8%)</td>
<td>3 (23%)</td>
<td>4 (15%)</td>
</tr>
<tr>
<td>rewritten</td>
<td>12 (92%)</td>
<td>10 (77%)</td>
<td>22 (85%)</td>
</tr>
</tbody>
</table>

As can be seen from Table 12.4, the rewritten texts were preferred in 85% of the cases.
The most common reasons for choosing the rewritten versions were:

- shorter sentences (R2 and R3)
- better paragraphing (R5)
- more direct with active voice (R16)
- fewer subordinated clauses (R4)
- better separation of instructions and descriptions (R12 and R13)
- clearer roles with active voice (R16)
- more natural ordering of items (R6)

The reasons for sometimes choosing the original texts were:

- sentences should begin with the important part (R6)
- active voice is unnecessarily commanding (R16)
- shorter sentences reduces the text flow (R2 and R3)
- active voice focuses too much on the application (R16)

As can be seen, some of the rules made the texts better as well as worse. However, all of these rules were significantly more often than not a reason for choosing the rewritten version. The exception is R6, which was about as beneficial as it was destructive.

We believe that a result of 85 % selection of our rewritten version is a good result, but the rule set can still be modified for the better. This is the purpose of Chapter 13.
Chapter 13

Rule modifications

When we studied our results from the tests of applicability and effects, we realized that some of our rules need to be slightly adjusted or that they did not function well in practice. In this chapter, we describe the changes that we made to our initial suggestions. The result is presented in Appendix B.

After the analysis was done, we realized that some of our suggested rules were never used; e.g., contractions may not be used (R7), articles may never be omitted (R9), particle verbs are not allowed (R18) and a sentence may not contain more than one negation (R20). It is probable that the suggested rules are not used because of the chosen text samples from IFS documentation.

13.1 Removed rules

We suggest removing four of our suggested rules; i.e., R2, R6, R9, and R19. The reason is the same for all these cases—our applicability study showed them all to be difficult to obey.

It makes sense to evaluate a text by using measures of averages, but it does not function well to enforce an average sentence length. When we studied the effects of our rules, we found that the rewriting because of the other rules resulted in the text conforming to this rule. In that sense, this rule does not have a positive effect and is therefore not needed. This, in combination with the difficulty associated with lengthening or shortening of sentences to affect the average just enough, is why we suggest to remove this rule.

As our applicability study showed, it is sometimes very difficult to decide in which order to write events in a sentence and if they always should be written in the order that they should occur. We believe that the ideas behind this rule are important. However, sometimes the writer might want to
emphasize the effect by putting it first and sometimes it is difficult to talk about causes and effects and natural order. Therefore, we suggest to remove this rule.

Our suggested rule about article omissions was hard to interpret. As shown in our applicability study the rule is either unnecessary because we require correct English before we apply the rules or it is difficult to speak of omissions. We therefore suggest to remove this rule.

Finally, our rule about ellipsis showed to be impossible to follow, since there does not seem to be a clear distinction of what is an ellipsis and what is not. Halliday and Hasan (1976) devote 80 pages to the phenomenon of ellipsis, which are very interesting, but show that it is a complicated matter. We agree that the writer should not leave out information that might be necessary to a non-expert, but we can not make this into a simple rule. Therefore, we suggest to remove this rule.

13.2 Modified rules

The rule about simple verb tenses (R10) required the most difficult rewritings according to our applicability test. In a few cases, it was very difficult to describe an event that happened in the past and that has ended. This is the reason for using perfect present tense and we find it necessary to allow it, even though it is not a simple tense. We understand the reasons for using only simple tenses, since it rules out some uncommon constructions such as will have been entered. However, we believe that the effort to write without perfect present tense without losing comprehensibility is far larger than the possible benefits. We therefore suggest to modify the rule to allow for the perfect present tense in addition to the simple tenses.

13.3 Tests of the new rule set

It would of course be appropriate to test our new set of controlled language rules; i.e., the rules after the suggested modifications. However, we have not performed such a test. The main reason for this is the limited time frame for this thesis work. But there are reasons to believe that the rule effects are not very different from those that we studied based on our initial rule suggestions.

The rule about average sentence length was in fact never used, since it was automatically fulfilled when we rewrote the texts. Neither did we ever use the rules about article omissions and ellipsis. The only different effects
come from the removal of the rule about event order (R6) and the addition of perfect present tense to allowed tenses (R10).

Free ordering of items permits the writer to put the important parts of a sentence first, which sometimes is preferred according to Gabrielsson (2005). However, some of the rewritings made because of this rule were appreciated.

Since perfect tense was not originally allowed, some sentences had to be rewritten in a way that made them longer without making them clearer, as expressed by Celinder (2005a). Allowing perfect present tense removes these problems.

It is our opinion that our new rule set is much easier to apply and we believe that the effects are slightly more positive in terms of comprehensibility and translatability compared to the originally suggested rule set.
Chapter 14

Implementation

We have developed our controlled language rules so that they would be easily applied by humans. Our applicability test has shown that it is fairly easy to detect when a rule is broken and to correct the problem. However, we believe that equipped with a controlled language tool, the authors will save time and effort.

This chapter describes our ideas for an implementation of our controlled language rules. We have developed an application that takes as input a piece of text and outputs whether the text complies a set of controlled language rules. If not, detailed information is given for each broken rule. This application can also be used as a plug-in to IFS applications development environment, which means that the text can be verified automatically when the author for instance writes a term definition. See Section 10.1 for more information about IFS term database.

14.1 IFS Development Environment

IFS has a complex development environment. The next core release of IFS Applications is developed within the Edge Project (Borg, 2005). A development client is used for several purposes. Among other things it contains the user interface connected to the IFS Term Database. In other words, this is the application where the authors enter term information.

The development client is called the Montgomery Client, since it belonged to the Montgomery Project. Now, this project has been merged into the Edge Project.

The Montgomery Client is written for the .NET platform with Microsoft Visual Studio.
14.2 Our CL checker

Our CL checker is developed with Microsoft Visual Studio using C# as programming language. We have split the program into three projects:

- a dynamically linked library (DLL) for checking a piece of text according to our CL rules
- a stand-alone sample application that checks a piece of text using the DLL mentioned above
- a modified version of the graphical user interface in IFS Term Database with added CL checking using the DLL mentioned above

We will continue this section with describing these three projects in detail. This section requires some understanding of object oriented programming concepts.

14.2.1 The checker DLL

A somewhat simplified view of our checker DLL is shown in Figure 14.1. In order to use our checker, the programmer first makes an instance of the Checker class. Then, in order to check a piece of text, this text can be sent as a string to the Check() method, which returns an object of the CheckerResult class.

When the checker is constructed, it searches for tests to perform on the text that it later will receive. All tests are classes that derive from the Test class. An example of this is TestMaxWordsPerSentence, which simply

![Figure 14.1: Our CL checker](image)
checks that our limit of 20 words per sentence (R3) is not exceeded. All tests have a Name and a Description and a method PerformTest() returning a TestResult that will be used by the checker when building a CheckerResult.

The implementation is by no means complete, but works as a proof of concept. In Section 14.3, we describe how our rules could be implemented in the future.

14.2.2 The sample application

![Figure 14.2: Our sample application](image)

The sample application is a graphical windows program that uses the DLL described above. It lets the user input some text, and when the user presses the "Perform Check" button, it simply sends the text to the Check() method of a Checker object as described above. The result is presented in a text box below. Figure 14.2 shows an example of how it might look.
14.2.3 The modified user interface

As mentioned in Section 10.2, IFS wanted to be able to perform controlled language checks on term definitions. To show that this is possible, we changed the program code for managing the term database user interface to include a label for displaying warnings when the text does not follow our rules. The design is shown in Figure 14.3.

14.3 Implementation ideas

In this section, we describe our thoughts concerning a future full implementation of a rule checker based on our CL rules. We go through our list of IFS CL rules one at a time; see Appendix B for the list of rules. This section could arguably be placed together with the other ideas for future work in Chapter 15, but since it belongs to the implementation and is quite comprehensive, we choose to put it here.

The list below is quite similar to the results of the text analysis in the rule applicability section; see Section 12.2.1. Note that the rules and their numbers have changed, since the implementation was done after the rule modifications presented in Chapter 13.
**R1** In order to check if a word exist in the IFS term database, all that is needed is a quick database lookup. However, since the term database does not contain common words, the checker will not be able to rule out words that are not in the term database today. A full controlled vocabulary is probably the easiest way of making this rule possible to check by a controlled language tool.

**R2** It is trivial to let the checker count words in a given sentence. But it is harder to tell what is a sentence, since they do not all end with periods. Also, periods can be used for abbreviations and capitals are used for proper names as well as first in a sentence etc. However, under normal conditions, it is easy to distinguish where a sentence begins and ends, and in the problematic cases the term database helps in that it can store the abbreviations with their periods.

**R3** Counting subordinated clauses requires syntactic knowledge. The easiest way to add support for this rule in the checker software would probably be to use libraries from an external syntactic tagger. A simpler solution is to cover the rule partially, by using hints from the text and present warnings to the user such as: “The sentence contains more than one comma, which indicates that it contains more than one subordinated clause. Consider splitting the sentence.” Apart from counting commas, the checker could look for words that are used to begin subordinated clauses; e.g., *which*.

**R4** Counting sentences in a paragraph is trivial when the definitions of sentences and paragraphs are clear. Sentences have already been discussed, and paragraphs are probably even easier to distinguish. In almost all cases where the user inserts a line-break, a new paragraph begins.

**R5** Since the number of commonly used contractions is limited, it is easy for the checker to store a list of those, and check the text against that list. Another approach could be to look for apostrophes that are not used for genitive, which can be simplified to all apostrophes that does not precede a single *s* or end a word after a *s, x,* or *z*.

**R6** Compounds would probably be quite easy to check, since a compound term with a specific meaning would be stored as a compound in the term database if it should be allowed. Other compound terms are still easy to locate, since their parts are in the term database. Formatting details such as hyphens or italics are also easy to extract and use by the checker.
R7 In order to examine verb tenses, an external tagger is necessary. However, it is still possible to for instance look for occurrences of *had*, which signals the disallowed past perfects or past perfect progressives.

R8 This rule states “unless there is a need to distinguish between the timing of events”. It is naturally impossible for the checker to determine whether the writer feels that such a need exists. However, the checker could alert the user and ask about constructions where other tenses are used than the simple present tense. But in order to analyze the verb tenses, an external syntactic tagger is most certainly needed.

R9 This rule requires a syntactic tagger to tell whether a verb is in the imperative mood. It also requires information regarding whether a text is an instruction or not, but this is probably evident from the type of text.

R10 This rule has the same requirements as R9.

R11 This rule makes sure that the author writes *you* instead of *the user*. The simplest way of checking this is probably to search for the phrase *the user*.

R12 In order to look for present participles, a syntactic tagger would be very beneficial. Without such an aid, it is still possible to alert the user for instance when words ending with *ing* are found, since these are likely to be present participles.

R13 This rule is hard for a checker to test. It requires syntactic information as well as semantic information that tells whether the participle should be interpreted as a description of a state or of an action. This can be partially solved by a dictionary. Microsoft Word uses such a dictionary, and does for instance say that *it is painted* is describing a state, but that *it is built* describes an action. In fact, both these sentences are ambiguous, so it is impossible for the checker to tell which is which without asking the user.

R14 This rule requires that the checker can access the term database and that it can determine if a word might be a gerund that is not included in the term database. The latter problem could be solved by a syntactic tagger. However, it is also possible to look for words with special endings such as *ing*, *ment*, and *tion*. Even if *ing*-words are often present participles, these are ruled out by R12, meaning that it does not matter whether they are participles or gerunds, since they are disallowed anyway, unless they are listed in the term database.
R15 The problem with phrasal verbs is that the same words often have other meaning if not pronounced as a phrasal verb. This ambiguity is the reason for disallowing them, but also makes it troublesome for the checker to determine what the author meant. One approach could be to use a dictionary with information about phrasal verbs and alert the user of all such constructions.

R16 Counting negations is very easy as the rule is formulated. It only requires a list of negations and a proper definition of a sentence for the checker to test this rule successfully.

As can be seen from the discussion above, a connection between the checker software and the term database is necessary. It is also evident that the use of an external tagger is needed in order to properly check the rules concerning syntax. A connection between the checker software and a dictionary could also be useful, if information such as phrasal verbs are not included in the term database in the future. As stated before, the term database has to be expanded to a full controlled vocabulary or combined with a list of allowed common words.
Part IV

Conclusions
Chapter 15

Future work

In this chapter, we present our ideas for further studies of controlled languages in software user documentation. The ideas that we have for extending our prototype controlled language checker has already been discussed in Section 14.3 and will not be addressed here.

15.1 Implementation of a full controlled vocabulary

We suggest a deeper investigation of the benefits of a controlled vocabulary with general words as well as terms. This makes checking easier and control tighter. This can be done as an extension to the term database or by using an external vocabulary.

15.2 Modifications to our controlled language ruleset

We believe that the rules that we suggest could be further modified and fine-tuned. As stated in Chapter 13, we had to remove the rule regarding ellipsis, because the area was too complex.

Other ideas for future rules are to study anaphora more closely and look at the problem that shorter sentences have a tendency to increase the amount of anaphora, which can be hard for machine translation. It would also be interesting to study nesting of phrases and not only clauses; prepositional phrases are often difficult to interpret correctly and would therefore be interesting to add to the controlled language rules discussion.
The difficulty of passive constructions most often lies in that the agent is unknown. However, it is possible to construct sentences with infinitive phrases that suffer from the same problem, but that are valid according to our controlled language rules. We therefore suggest to study infinitive phrases as well and see if it is possible to better control their use.

Our numerical limits deserves to be tested more thoroughly as well. We believe that sentence length limitation is good, but we have no proof that 20 words is not the optimum limit. We have the impression that this has primarily been studied from the readability viewpoint. It would be interesting to study the effects on translatability and re-usability, because—as we mentioned briefly in Section 11.2—shorter sentences can lead to increased use of anaphora which complicates this.

15.3 Choice of a CL tool for IFS

In our thesis, we do not recommend the use of any existing CL checker but we do give a description of the features of the existing tools. It is interesting for IFS to investigate whether it is better to use an already existing tool during the process of writing documentation or to develop a tool of their own. A further investigation of the functionality of already existing tools is needed in order to decide if they match the needs of IFS.

15.4 Further testing of comprehensibility and translatability

Measuring and testing comprehensibility, readability, and translatability is a large area to study. The results of the application of a controlled language on IFS texts could be further tested by using other measures than the ones that we used. Also, a deeper analysis of the results of our interviews could be interesting. This could increase the reliability of our conclusions and possibly give even more proof that controlled language rules indeed improve comprehensibility and translatability. The tests could be done on the rules individually, on combinations of rules, or on all of the rules together.

We chose to test our controlled language rules with help from technical writers, who could use these rules during the creation of the documentation. A further comprehensibility investigation would be interesting to perform from the end-user perspective, the translators and also from programmers’ point of view, who are often the authors of the end-user documentation.
In Section 3.9.3, we mention the Bleu method for measuring translation quality. We intended to use this measure in our applicability study of our rules. We asked technical writers at IFS to rewrite parts of the original texts according to our rules, and we intended to compare their rewritings to those performed by us. However, we never got any texts back from these employees. We still think it is interesting to view the rewriting process as a kind of translation and to use measures of translatability as a way of testing rule applicability.
Early in our study, we came across an article written by Nyberg et al. (2003) about the many advantages of controlled languages for organizations that write and maintain complex technical manuals. This article made us realize that the documentation issues that IFS was dealing with were common. Some of the issues that IFS faced and that Nyberg et al. (2003) discussed were:

- terminology and sentence structure inconsistency because of lack of standardization\(^1\)
- how to achieve uniformity of writing style, so that text written for one part of IFS documentation can be reused elsewhere when appropriate
- how to use translation aids (such as translation memory tools) more effectively and decrease cost and time.

Controlled languages intend to improve readability, text comprehensibility, and translatability of the text (Nyberg et al., 2003; Adriaens and Schreurs, 1992). This has been the main reason for introducing CL in industry before (Wojcik and Hoard, 1996). The goal of IFS with giving us the opportunity to write this thesis was to help them reduce time and cost for documentation work and translation. The goals of IFS match the goals of companies that have introduced CL in the past.

In order to measure comprehensibility and translatability for IFS texts, we used simple measures such as readability index (Flesh Reading Ease and Flesh-Kincaid Grade Level), the nominal quotient and assessment tests. There are many comprehensibility and translatability measures appropriate

\(^1\)Today the IFS source texts are greatly varied due to fact that the documentation is written in different places all over the world by many different writers.
to use; see Section 2.4, Section 2.5 and Section 3.9 for more information about these.

As we mentioned in Section 5.1, the main ingredients in a controlled language are a vocabulary and a set of grammatical rules. IFS is currently working with the vocabulary by developing a term database which is a step on the way toward a full vocabulary for a controlled language. To construct a full vocabulary IFS needs to add common words to its term database; see Section 10.2 for a further discussion of this. The other component of a controlled language, a set of grammatical rules, is where most of the focus of our thesis lies. Based on our case study we have constructed a full set of controlled language rules that can be added to the normal writing rules of American English. The controlled language rules that we propose are applicable and useful to IFS.

The main problems encountered in comprehensibility and translatability are sentence length, sentence complexity, and terminology; see Chapter 9.6 for more details about this. Every rule that we suggest aims at solving some of the problems at IFS concerning comprehensibility and translatability. For instance, our rule about limiting sentence length solves one of IFS major problems—sentence limitation—and the limitation of subordinated clauses solves another one of IFS problems—the complexity problem.

The list of language rules which we put together can help writers at IFS during the documentation process; see Appendix B.

Our case study shows that readability, comprehensibility, and translatability of IFS texts written in a controlled language are improved. We showed this by interviewing experienced technical writers at IFS, who in the majority of cases preferred IFS texts written in a controlled language to the original IFS texts when asked to choose between the two. Our case study at IFS contributes to the knowledge in the area of controlled languages by showing that the introduction of a CL in a software company like IFS helps to improve readability, comprehensibility, and translatability of the company’s software user documentation.

The CL rules we developed are formulated very generally and can easily be used all over the software domain. Most of our rules give restrictions on language that promote readability, comprehensibility, and translatability in general and are based on existing knowledge in these areas. A few examples of our general restrictions are only allowing standardized terminology, limiting the sentence length and complexity, limiting the paragraph length, and only allowing the simple verb tenses.

One of the primary goals of IFS is saving money and time for documentation and translation work. A natural question at this point is if IFS is planning to introduce CL in the nearby future. They would gain a lot by do-
ing so; e.g., improving the quality of documentation, reducing time to create new documentation and making it easy to reuse documentation and translations. A result of the standardization work with the terminology at IFS and the creation of a term database, is that the company is saving 4,500 hours of documentation work from IFS countries during 2005 and SEK 3 million in application translations savings (Celinder, 2005d).

Currently, IFS does not have the resources to introduce a controlled language but they are planning on doing so in the future. This is partly because our thesis helped them realize that there are many benefits to be gained. We hope that IFS will use our language rules by updating their existing Style and Writing Guide to include our rules. Hopefully, IFS will in the future use our rules as a starting point for their own controlled language. We wish IFS the best of luck in their future work within the exciting controlled language field.
Bibliography


Altwarg, R. (2000). Controlled languages: An introduction. Macquarie University Graduate Program in Speech and Language Process-


---

\(^2\)Power Point is a registered trademark of Microsoft Corporation.

Hammarström, P. (2004b). Translation and user support objectives. IFS internal document (Published on IFS WIS).


IFS (2004c). IFS style and writing guide. IFS internal document (Published on IFS WIS).


Appendix A

Summary of suggested rules

1. Only terms that are approved in the IFS Term Database are allowed.
2. The average sentence length should be 13–17 words.
3. All sentences should contain no more than 20 words.
4. A maximum of one subordinated clause is allowed in a sentence.
5. All paragraphs should contain no more than 6 sentences.
6. Always write events in the order that they are supposed to occur.
7. Contractions may not be used.
8. Compound terms may not consist of more than three words.

   **Exception A:** The rule does not apply if the entire compound is an approved term in the IFS Term Database

   **Exception B:** The rule does not apply if the compound consists of words that are visually grouped together into units and there are no more than three such units. Each unit must itself meet the requirements of a compound.

9. Articles (a, an, the, this, these) may never be omitted.
10. Use only the simple verb tenses.
11. Use simple present tense unless there is a need to distinguish between the timing of events.
12. Instructions should be written using the imperative mood.
13. Descriptions should be written using the indicative mood.

14. Descriptions should be written in second person form when there is a need to refer to the user.

15. Present participles are not allowed.

16. Past participles are allowed only when describing a state and either precedes the noun it determines or follows a form of to be or become.

17. Gerunds are not allowed unless explicitly listed as terms in the IFS Term Database.

18. Phrasal verbs are not allowed.

19. Elliptical constructions are not allowed.

20. A sentence may not contain more than one negation; e.g., no, not, nothing, neither, never.
Appendix B

Summary of IFS CL rules

1. Only terms that are approved in the IFS Term Database are allowed.

2. All sentences should contain no more than 20 words.

3. A maximum of one subordinated clause is allowed in a sentence.

4. All paragraphs should contain no more than 6 sentences.

5. Contractions may not be used.

6. Compound terms may not consist of more than three words.
   
   **Exception A:** The rule does not apply if the entire compound is an approved term in the IFS Term Database
   
   **Exception B:** The rule does not apply if the compound consist of words that are visually grouped together into units and there are no more than three such units. Each unit must itself meet the requirements of a compound.

7. Use only the simple verb tenses or the perfect present tense.

8. Use simple present tense unless there is a need to distinguish between the timing of events.

9. Instructions should be written using the imperative mood.

10. Descriptions should be written using the indicative mood.

11. Descriptions should be written in second person form when there is a need to refer to the user.
12. Present participles are not allowed.

13. Past participles are allowed only when describing a state and either precedes the noun it determines or follows a form of to be or become.

14. Gerunds are not allowed unless explicitly listed as terms in the IFS Term Database.

15. Phrasal verbs are not allowed.

16. A sentence may not contain more than one negation; e.g., no, not, nothing, neither, never.
Appendix C

Measuring information density

In this appendix, we explain how we did measure the information density in a paragraph of the original IFS text and in our corresponding rewritten version. The text that we used was the first paragraph from Vertex Integration.

C.1 NQ analysis of the original version

Each word cluster in Table C.1 is counted as one entity. As mentioned in Section 2.3.3 information density is calculated according to the following formula:

\[ NQ = \frac{nouns + prepositions + participles}{pronouns + adverbs + verbs} \]

The following notations are used in the tables below: n–noun, v–verb, prep–preposition, adv–adverb, pro–pronoun and part–participle.

This leads to the result that the information density for the original text is 2.91 in this paragraph.

\[ NQ = \frac{64}{22} \approx 2.91 \]

C.2 NQ analysis of the rewritten version

The word clusters in Table C.3 are counted as one entity just as in the previous calculation.

The resulting information density for the rewritten text is 2.39 in this paragraph.

\[ NQ = \frac{55}{23} \approx 2.39 \]
<table>
<thead>
<tr>
<th>Word</th>
<th>Part-of-speech</th>
<th>Word</th>
<th>Part-of-speech</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vertex Inc.</td>
<td>n</td>
<td>Vertex tax calculation</td>
<td>n</td>
</tr>
<tr>
<td>third-party software solution</td>
<td>n</td>
<td>based part</td>
<td>part</td>
</tr>
<tr>
<td>for</td>
<td>prep</td>
<td>on prep</td>
<td></td>
</tr>
<tr>
<td>markets</td>
<td>n</td>
<td>nine-digit proprietary GeoCode</td>
<td>n</td>
</tr>
<tr>
<td>called</td>
<td>part</td>
<td>code</td>
<td>n</td>
</tr>
<tr>
<td>Quantum for Sales and Use Tax</td>
<td>n</td>
<td>combination</td>
<td>n</td>
</tr>
<tr>
<td>sales and use taxes</td>
<td>n</td>
<td>of prep</td>
<td></td>
</tr>
<tr>
<td>Taxes</td>
<td>n</td>
<td>taxation entities</td>
<td>n</td>
</tr>
<tr>
<td>calculated</td>
<td>part</td>
<td>for</td>
<td>n</td>
</tr>
<tr>
<td>based</td>
<td>part</td>
<td>state</td>
<td>n</td>
</tr>
<tr>
<td>on</td>
<td>prep</td>
<td>county</td>
<td>n</td>
</tr>
<tr>
<td>state</td>
<td>n</td>
<td>city</td>
<td>n</td>
</tr>
<tr>
<td>county</td>
<td>n</td>
<td>digits</td>
<td>n</td>
</tr>
<tr>
<td>city</td>
<td>n</td>
<td>state</td>
<td>n</td>
</tr>
<tr>
<td>postal (zip) codes</td>
<td>n</td>
<td>county</td>
<td>n</td>
</tr>
<tr>
<td>entered</td>
<td>part</td>
<td>digit</td>
<td>n</td>
</tr>
<tr>
<td>in</td>
<td>prep</td>
<td>city</td>
<td>n</td>
</tr>
<tr>
<td>customer order address fields</td>
<td>n</td>
<td>code</td>
<td>n</td>
</tr>
<tr>
<td>Through</td>
<td>prep</td>
<td>displayed</td>
<td>part</td>
</tr>
<tr>
<td>subscription service</td>
<td>n</td>
<td>in</td>
<td>prep</td>
</tr>
<tr>
<td>Vertex</td>
<td>n</td>
<td>Jurisdiction Code field</td>
<td>n</td>
</tr>
<tr>
<td>taxation entities</td>
<td>n</td>
<td>IFS Applications windows</td>
<td>n</td>
</tr>
<tr>
<td>percentages</td>
<td>n</td>
<td>In City check box</td>
<td>n</td>
</tr>
<tr>
<td>on</td>
<td>prep</td>
<td>used</td>
<td>n</td>
</tr>
<tr>
<td>basis</td>
<td>n</td>
<td>address</td>
<td>n</td>
</tr>
<tr>
<td>process</td>
<td>n</td>
<td>within</td>
<td>n</td>
</tr>
<tr>
<td>of</td>
<td>prep</td>
<td>city limits</td>
<td>n</td>
</tr>
<tr>
<td>calculating</td>
<td>part</td>
<td>value</td>
<td>n</td>
</tr>
<tr>
<td>taxes</td>
<td>n</td>
<td>used</td>
<td>part</td>
</tr>
<tr>
<td>for</td>
<td>prep</td>
<td>to prep</td>
<td></td>
</tr>
<tr>
<td>sales</td>
<td>n</td>
<td>city taxes</td>
<td>n</td>
</tr>
</tbody>
</table>
Table C.2: Pronouns, adverbs, and verbs in the original version

<table>
<thead>
<tr>
<th>Word</th>
<th>Part-of-speech</th>
</tr>
</thead>
<tbody>
<tr>
<td>provides</td>
<td>v</td>
</tr>
<tr>
<td>which</td>
<td>adv</td>
</tr>
<tr>
<td>calculates</td>
<td>v</td>
</tr>
<tr>
<td>are</td>
<td>v</td>
</tr>
<tr>
<td>updates</td>
<td>v</td>
</tr>
<tr>
<td>This</td>
<td>adv</td>
</tr>
<tr>
<td>greatly</td>
<td>adv</td>
</tr>
<tr>
<td>simplifies</td>
<td>v</td>
</tr>
<tr>
<td>correctly</td>
<td>adv</td>
</tr>
<tr>
<td>represents</td>
<td>v</td>
</tr>
<tr>
<td>identify</td>
<td>v</td>
</tr>
<tr>
<td>identify</td>
<td>v</td>
</tr>
<tr>
<td>is</td>
<td>v</td>
</tr>
<tr>
<td>is</td>
<td>v</td>
</tr>
<tr>
<td>indicate</td>
<td>v</td>
</tr>
<tr>
<td>is</td>
<td>v</td>
</tr>
<tr>
<td>This</td>
<td>adv</td>
</tr>
<tr>
<td>will</td>
<td>v</td>
</tr>
<tr>
<td>be</td>
<td>v</td>
</tr>
<tr>
<td>determine</td>
<td>v</td>
</tr>
<tr>
<td>calculate</td>
<td>v</td>
</tr>
<tr>
<td>Word</td>
<td>Part-of-speech</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Vertex Inc.</td>
<td>n</td>
</tr>
<tr>
<td>third-party</td>
<td>n</td>
</tr>
<tr>
<td>software solution</td>
<td>n</td>
</tr>
<tr>
<td>for</td>
<td>prep</td>
</tr>
<tr>
<td>markets</td>
<td>n</td>
</tr>
<tr>
<td>proprietary GeoCode</td>
<td>n</td>
</tr>
<tr>
<td>called</td>
<td>part</td>
</tr>
<tr>
<td>Quantum for Sales and Use Tax</td>
<td>n</td>
</tr>
<tr>
<td>sales and use taxes</td>
<td>n</td>
</tr>
<tr>
<td>tax calculation</td>
<td>n</td>
</tr>
<tr>
<td>information</td>
<td>n</td>
</tr>
<tr>
<td>about</td>
<td>prep</td>
</tr>
<tr>
<td>state</td>
<td>n</td>
</tr>
<tr>
<td>county</td>
<td>n</td>
</tr>
<tr>
<td>city</td>
<td>n</td>
</tr>
<tr>
<td>postal codes</td>
<td>n</td>
</tr>
<tr>
<td>from</td>
<td>prep</td>
</tr>
<tr>
<td>customer order</td>
<td>n</td>
</tr>
<tr>
<td>address fields</td>
<td>n</td>
</tr>
<tr>
<td>Through</td>
<td>prep</td>
</tr>
<tr>
<td>subscription service</td>
<td>n</td>
</tr>
<tr>
<td>Vertex</td>
<td>n</td>
</tr>
<tr>
<td>taxation entities</td>
<td>n</td>
</tr>
<tr>
<td>percentages</td>
<td>n</td>
</tr>
<tr>
<td>on</td>
<td>prep</td>
</tr>
<tr>
<td>windows</td>
<td></td>
</tr>
<tr>
<td>basis</td>
<td>n</td>
</tr>
<tr>
<td>taxes</td>
<td>n</td>
</tr>
<tr>
<td>for</td>
<td>prep</td>
</tr>
</tbody>
</table>
Table C.4: Pronouns, adverbs, and verbs in the rewritten version

<table>
<thead>
<tr>
<th>Word</th>
<th>Part-of-speech</th>
</tr>
</thead>
<tbody>
<tr>
<td>provides</td>
<td>v</td>
</tr>
<tr>
<td>This</td>
<td>pro</td>
</tr>
<tr>
<td>is</td>
<td>v</td>
</tr>
<tr>
<td>it</td>
<td>pro</td>
</tr>
<tr>
<td>calculates</td>
<td>v</td>
</tr>
<tr>
<td>uses</td>
<td>v</td>
</tr>
<tr>
<td>updates</td>
<td>v</td>
</tr>
<tr>
<td>This</td>
<td>pro</td>
</tr>
<tr>
<td>makes</td>
<td>v</td>
</tr>
<tr>
<td>it</td>
<td>pro</td>
</tr>
<tr>
<td>calculate</td>
<td>v</td>
</tr>
<tr>
<td>correctly</td>
<td>v</td>
</tr>
<tr>
<td>is</td>
<td>v</td>
</tr>
<tr>
<td>represents</td>
<td>v</td>
</tr>
<tr>
<td>identify</td>
<td>v</td>
</tr>
<tr>
<td>identify</td>
<td>v</td>
</tr>
<tr>
<td>identify</td>
<td>v</td>
</tr>
<tr>
<td>displays</td>
<td>v</td>
</tr>
<tr>
<td>indicate</td>
<td>v</td>
</tr>
<tr>
<td>is</td>
<td>v</td>
</tr>
<tr>
<td>This</td>
<td>pro</td>
</tr>
<tr>
<td>determines</td>
<td>v</td>
</tr>
<tr>
<td>calculate</td>
<td>v</td>
</tr>
</tbody>
</table>
Appendix D

Sample texts

In this appendix, we present the three texts that we have used when testing our suggested rules. They are gathered from the documentation at IFS and they are taken from the three most common types of documentation texts at IFS; see Section 9.3.

In each section below, we first present the original text with all sentences numbered. Then, we go through the analysis of the text sentence by sentence and we make a paragraph and text analysis. We thereafter present the rewritten version that we suggest. This is the result of a process where we rewrote the texts individually and then compared our texts and agreed on the best rewriting for each sentence, resulting in a merged version that we both liked. Finally, we give a short comment on our version.

D.1 Vertex Integration

This is an About Description written in Sweden in the Financials domain.

D.1.1 Original text

Vertex Integration

1Vertex Inc. provides a third-party software solution for the U.S. and Canadian markets called Quantum for Sales and Use Tax, which calculates sales and use taxes. 2Taxes are calculated based on the state, county, city, and postal (zip) codes entered in customer order address fields. 3Through a subscription service, Vertex updates the taxation entities and percentages on a monthly basis. 4This greatly simplifies the process of correctly calculating taxes for sales. 5Vertex tax calculation is based on a nine-digit proprietary GeoCode. 6Each code represents a combination of taxation entities for state,
county, and city. The first two digits identify the state, the next three identify the specific county, and the last four digits identify the city. This code is displayed in the Jurisdiction Code field in several IFS Applications windows. The In City check box is used to indicate whether or not the address is within the city limits. This value will be used to determine whether or not to calculate city taxes.

11 To enable tax calculation using the Quantum software, you must define the sales tax calculation method at the company level. In the Enterprise/Company/Company/Invoice tab, select Quantum for Sales and Use Tax from the Sales Tax Calculation Method list.

12 For tax calculation to work accurately with Vertex integration, it is important to enter the correct address information. Values for all the U.S. states are pre-installed in Enterprise/General Enterprise/State Codes. Use the List of Values whenever possible to select states for the state address fields. County, city, and postal (zip) codes must be entered manually. Generic tax codes are used to enable Vertex to retrieve the appropriate tax amounts and percentages. These tax codes are established in Accounting Rules/Tax Codes/Tax Codes, and can be selected in various windows via a List of Values. Do not enter tax percentages if you have Vertex integration installed; the system will automatically enter the percentages and calculate the tax amounts.

13 Any time you make changes to any of the required address values and save the record, Vertex will use that new data to determine the GeoCode, which will be reflected in a changed value in the Jurisdiction Code field. If values in any of the mandatory address fields are missing or misspelled, you will receive an error message indicating that the address information is incomplete and that tax calculations cannot be performed. Valid address fields are needed for correct Geo Code retrieval.

D.1.2 Analysis of the original text

The 22 sentences are analyzed with the following result:

1. R3: 26 words
   R16: called is not preceded by a form of “to be” or “become”

2. R16: are calculated describes an action

3. OK

4. R17: calculating is a gerund
5. OK

6. OK

7. R3: 22 words

8. R16: *is displayed* is not describing a state

9. R16: *is used* is not describing a state

10. R11: unnecessary use of future tense (*will be used*)
    R16: *be used* is not describing a state

11. R6: *to enable* proceeds its requirement *must define*
    R15: *using* is a present participle

12. OK

13. R6: *enter address* are supposed to occur before *work accurately* but is placed after

14. OK

15. OK

16. R16: *be entered* is not describing a state

17. R16: *are used* is not describing a state

18. R3: 24 words
    R16: *are established* is not describing a state

19. R3: 23 words

20. R3: 39 words
    R4: 2 subordinated clauses
    R16: *be reflected* is not describing a state
    R16: *changed* is not describing a state

21. R3: 33 words
    R15: *missing* is a present participle
    R16: *be performed* is not describing a state

22. R16: *are needed* is not functioning as an adjective
The four paragraphs are analyzed with regard to number of sentences. Paragraph 1 (sentences 1–10) and paragraph 3 (sentences 13–19) both violate R5.

Finally, the text as a whole is analyzed. The text has an average sentence length of 18.4 words, which violates R2.

D.1.3 Rewritten text

**Vertex Integration**

1. Vertex Inc. provides a third-party software solution for the U.S. and Canadian markets. This solution is called Quantum for Sales and Use Tax. It calculates sales and use taxes.

2. The tax calculation uses information about state, county, city, and postal codes from the customer order address fields.

3. Through a subscription service, Vertex updates the taxation entities and percentages on a monthly basis. This makes it simpler to calculate taxes for sales correctly.

4. The Vertex tax calculation is based on a nine-digit proprietary GeoCode. Each code represents a combination of taxation entities for state, county, and city.

5. The first two digits identify the state, and the next three digits identify the specific county. The last four digits identify the city.

6. The Jurisdiction Code field displays this code in several IFS Applications windows.

7. The In City check box indicates whether the address is within the city limits. This value determines whether to calculate city taxes.

8. You must define the sales tax calculation method at the company level to enable the Quantum Software to calculate taxes.


10. It is important to enter the correct address information for the tax calculation to work accurately with Vertex integration.

11. Values for all the U.S. states are pre-installed in Enterprise/General Enterprise/State Codes.

12. Use the List of Values whenever possible to select states for the state address fields.

13. You must enter county, city, and postal (zip) codes manually.

14. Vertex uses generic tax codes to enable the retrieval of the appropriate tax amounts and percentages.

15. You can establish these tax codes in Accounting/Rules/Tax Codes/Tax Codes. Then, you can select them in various windows via a List of Values.

16. Do not enter tax percentages if you have Vertex integration installed. The system automatically enters the percentages and calculates the tax amounts.
20 When you change a required address value and save the record, Vertex uses that new data to determine the GeoCode. This changes the value in the Jurisdiction Code field. 21 If any of the mandatory address fields is empty or has a misspelled value, you will receive an error message. This error message indicates that the address information is incomplete and that the system cannot calculate taxes. 22 Vertex requires that the address information is valid in order to retrieve Geo Codes correctly.

D.1.4 Analysis of the rewritten text

When we rewrote the text, we paid attention to all rules on sentence level and paragraph level. What remains is the text level. Only R2 is such a rule. The rewritten text has an average of 13.8 words per sentence and the rule is thereby followed.

D.2 Create Company In All Components

This is an Activity Description written in Sweden in the Financials domain.

D.2.1 Original text

Create Company In All Components

Explanation

1 This is the main activity in the Create/Update company process. 2 All data defined during the previous activities will now be sent from the client to the business logic server, and processed by all components supporting the process in order to create basic data for the company.

Prerequisites

- 3 Company name and type of company must have been previously defined.
- 4 Type of source, template or company, must have been previously defined.
- 5 If you need to create language translations for the company, the languages must have been previously defined. 6 Detailed data for the company must have been previously defined.
The activity must be triggered either when creating a new company or when updating an existing company.

System Effects

A company will be created as a result of this activity. The processing is performed according to the following:

1. Find all components that support the Create Company process.
2. Also find out if the component supports company creation from template/company or if it is a component version from an old IFS Applications baseline that only supports the process using so-called default data that is independent of the selected source template or company. For more information, see the help file associated with the Component Registration window in IFS/Enterprise.

3. All active Logical Units in all active components are accessed through a standard interface. This ensures that all data related to the Logical Unit is retrieved from the source (a company template or existing company) and then applied to the company that is being created or updated.

4. Language translations are handled as follows:
   - When a new company is created, the translations selected from the source (company or template) are created in the new company for all active Logical Units.
   - When an existing company is updated, only the language translations that are used by that company will be updated. The updated language translations come from the source company.

"Note: The Update Company activity only adds basic data (and corresponding translations) to the current company.

5. All errors during the process are logged.

6. The process ends after having accessed all active Logical Units in all active components.

It is important to point out that this activity will never fail, i.e., if errors are found, then these errors are logged and the activity will continue. This means that the company will be created at least in IFS/Enterprise.
It is also important to point out that updating a company always means that basic data and translations are added. Update Company will never remove or update any data in the current company.

D.2.2 Analysis of the original text

The 25 sentences are analyzed with the following result:

1. OK

2. R3: 36 words
   R11: unnecessary use of future tense (will)
   R15: supporting is a present participle
   R16: defined is not acting as an adjective
   R16: sent is not describing a state
   R16: processed is not describing a state

3. R10: have been defined is not a simple verb tense
   R16: been defined is not describing a state

4. R10: have been defined is not a simple verb tense
   R16: been defined is not describing a state

5. This sentence requires rewriting beyond the scope of our rules. We therefore omit this sentence from the analysis and rewriting.

6. This sentence requires rewriting beyond the scope of our rules. We therefore omit this sentence from the analysis and rewriting.

7. This sentence requires rewriting beyond the scope of our rules. We therefore omit this sentence from the analysis and rewriting.

8. R16: be created in not describing a state

9. R17: processing is a gerund
   R15: following is a present participle
   R16: is performed is not describing a state

10. R13: The sentence is descriptive but written using the imperative mood.

11. R3: 44 words
    R4: 4 subordinated clauses
    R13: The sentence is descriptive but written using the imperative
mood.

12. R16: *associated* is not preceded by a form of to be or become

13. R16: *are accessed* is not describing a state

14. R3: 33 words
   R4: 2 subordinated clauses
   R16: *related* is not preceded by a form of “to be” or “become”
   R16: *is retrieved* is not describing a state
   R16: *applied* is not describing a state
   R16: *created* is not describing a state
   R16: *updated* is not describing a state

15. R16: *are handled* is not describing a state

16. R3: 26 words
   R16: *is created* is not describing a state
   R16: *selected* is not preceded by a form of to be or become
   R16: *are created* is not describing a state

17. R11: unnecessary use of future tense (*will*)
   R16: *is updated* is not describing a state
   R16: *are used* is not describing a state

18. OK

19. R15: *corresponding* is a present participle

20. R16: *is logged* is not describing a state

21. R6: *ends* precedes its cause: *having accessed*
   R6: the sentence belongs to the step-by-step description but is placed after the general notes
   R10: *having accessed* is not a simple verb tense

22. R3: 27 words
   R11: unnecessary use of future tense (*will fail*)
   R11: unnecessary use of future tense (*will continue*)
   R16: *are found* is not describing a state
   R16: *are logged* is not describing a state

23. R11: unnecessary use of future tense (*will*)
   R16: *created* is not describing a state
24. R4: 2 subordinated clauses
   R16: *are added* is not describing a state
   R17: *updating* is a gerund

25. R11: unnecessary use of future tense (*will*)

All of the paragraphs are short enough to obey R5. The average sentence length in the text is 17.6 words, which violates R2.

### D.2.3 Rewritten text

**Create Company in All Components**

**Explanation**

1. This is the main activity in the Create/Update company process. 2. This activity sends all data in the previous activities from the client to the business logic server. All components that support the process then handle the data in order to create basic data for the company.

**Prerequisites**

3. You must first define the Company Name and Company Type.
4. You must first select Template or Company as Type of Source.
5-7 ignored

**System Effects**

8-9. This activity creates a company by these steps:

- 10. The system finds all components that support the Create Company process. 11. In addition, the system examines each component to investigate if it supports company creation from Template or Company. Components from old IFS Application baselines only support default data, which is independent of the selected Type of Source. 12. For more information, see the help file for the Component Registration window in IFS/Enterprise.

- 13. The system accesses all active Logical Units in all active components through a standard interface. 14. This ensures that the system retrieves all data for the Logical Unit from the selected source template or company. The system then applies this data to the company that it creates or updates.
• The system ends the process.

The system handles language translations as follows:

15 When you create a company

The system uses the translations that you selected to create new translations in the company for all active Logical Units.

17 When you update an existing company

The system only updates the language translations that the company uses. The updated language translations come from the source company.

19 Note: The Update Company Activity only adds basic data (and its translations) to the current company.

21 The system logs all errors during the process.

22 It is important to point out that this activity never fails. If the system finds errors, it logs them and the activity continues. This means that the process creates the company at least in IFS/Enterprise.

24 It is also important to point out what happens when the system updates the company. This always means that the system adds basic data and translations to the company. Update Company never removes or updates any data in the current company.

D.2.4 Analysis of the rewritten text

After the text has been rewritten to follow all sentence and paragraph rules, the average number of words per sentence is 13.5, which means that the text obeys R2.

D.3 Automatic Sourcing Candidates

This is a Window Description written in Sri Lanka in the Distribution domain.

D.3.1 Source text

Automatic Sourcing Candidates

Usage

1 Use this window to view all sourcing alternatives that the automatic sourcing process has used in its search for the best alternative. 2 These sourcing
alternatives originate from the source set entered on the sourcing rule. ³The sourcing rule is connected to the sales part alone, to the customer and sales part, or to the customer address and sales part.

⁴The candidates shown in this dialog box are the sourcing alternatives that were equally good; they were separated through the sourcing objectives, i.e., the criteria. ⁵The Selected check box is selected on the chosen alternatives. ⁶For instance: five sourcing alternatives out of seven can deliver the whole sales quantity in time. ⁷All five will be shown in the dialog. ⁸However, two of the alternatives have the same total shipping time (we assume that only shipping time is used as a criterion). ⁹These two will have the Selected check box selected. ¹⁰The automatic sourcing process will pick the first alternative of the two to supply the customer.

¹¹A sourcing rule consists of a source set and sourcing objectives. ¹²The source set is a set of all sourcing alternatives. ¹³Sourcing alternatives can be external suppliers, internal suppliers, in-house manufacturing, or company inventory. ¹⁴The automatic sourcing will select the best alternative from this set. ¹⁵The best alternative will determine the supply code and, when appropriate, the supplier on the customer order line. ¹⁶The sourcing objectives consist of a sequence of criteria that the automatic sourcing will use when more than one sourcing alternative is suited to supply the customer order line. ¹⁷The following algorithm is used when the automatic sourcing selects the best sourcing alternative.

1. ¹⁸From the source set, find all sourcing alternatives that can deliver the whole sales quantity in time.

2. ¹⁹If more than one sourcing alternative is found in step 1, use the criteria in the sourcing objective to narrow down the possible alternatives.

3. ²⁰If more than one sourcing alternative remains after step 2, pick the first alternative.

4. ²¹If no sourcing alternative was found in step 1, find the alternative from the source set that can deliver the whole order quantity earliest.

5. ²²If more than one sourcing alternative can deliver on the same earliest possible delivery date, use the criteria in the sourcing objective to narrow down the possible alternatives.

6. ²³If more than one sourcing alternative remains after step 5, pick the first alternative.
Fields

[Field descriptions without at least one full sentence are omitted. Only full sentences are numbered and analyzed]

**Supplier:** The supplier selected by the automatic sourcing process. 24 This can be an internal supplier or an external supplier. 25 If this field is null, the order will be delivered from the site taking the order, i.e. the site on the order line.

**Supply Site:** The supply site for this sourcing alternative. 26 If this field is null and the Supplier field has a value, the supplier is an external supplier.

**Selected:** The check box that indicates whether this sourcing alternative is selected. 27 All records that were considered good alternatives during the final selection will have this check box selected. 28 If several sourcing alternatives are selected, it means that these alternatives were equally good.

**Ship Via Code:** The default ship via code for this sourcing alternative. 29 Depending on the supply code, this ship via code represents a direct or a transit delivery.

**External Transport Lead Time:** The external transport lead time when shipping parts. 30 The external transport lead time is expressed in calendar days, which includes weekends and holidays. 31 This value is a key parameter for decision making during the sourcing process, and becomes a part of the total lead time when the delivery dates for the orders are calculated.

**Earliest Possible Delivery Date:** The earliest possible delivery date, i.e. the first possible date when the end customer can receive the whole sales quantity from this sourcing alternative. 32 This date is based on an Available To Promise (ATP) analysis only if the inventory part, on the appropriate site, has the availability or the online consumption check option selected. 33 Consequently, this means that in other cases this date is only based on lead time calculations.

**ATP (Plannable Qty):** The ATP at this sourcing alternative, i.e. this is the quantity that could be delivered in time in order to fulfill the end customer’s desired delivery date. 34 If this quantity is less than the revised quantity, the earliest possible delivery date will be later than
the desired delivery date. ATP is only analyzed if the inventory part, on the appropriate site, has the availability or the online consumption check option selected.

**ATP (Available To Promise Date):** The first date when full quantity can be picked from inventory to be delivered to the end customer. This field only displays a value if an ATP analysis is done.

### D.3.2 Analysis of the source text

The 36 sentences are analyzed with the following result:

1. R3: 22 words
   - R10: *has used* is not a simple verb form
   - R13: The sentence is descriptive but written using the imperative mood

2. R16: *entered* is not preceded by a form of “to be” or “become”

3. R3: 24 words

4. R3: 25 words
   - R16: *shown* is not preceded by a form of to be or become

5. OK

6. OK

7. R16: *be shown* is not describing a state

8. R3: 22 words
   - R6: The assumption should precede the sentence it explains
   - R16: *is used* is not describing a state

9. R16: *selected* is not processed by a form of to be or become

10. OK

11. OK

12. OK

13. OK

14. OK

15. OK
16. R3: 29 words
17. R16: *is used* is not describing a state
18. OK
19. R3: 24 words
   R16: *is found* is not describing a state
20. OK
21. R3: 24 words
   R16: *was found* is not describing a state
22. R3: 28 words
23. OK
24. OK
25. R3: 24 words
   R15: *taking* is a present participle
   R16: *be delivered* is not describing a state
26. OK
27. R11: unnecessary use of future tense (*will have*)
   R16: *were considered* is not describing a state
   R16: *selected* is not preceded by a form of “to be” or “become”
28. OK
29. R15: *depending* is a present participle
30. OK
31. R3: 31 words
   R16: *are calculated* is not describing a state
   R17: *decision making* is a gerund
32. R3: 30 words
   R16: *selected* is not preceded by a form of “to be” or “become”
33. OK
34. R3: 22 words
35. R3: 22 words
   R16: *is analyzed* is not describing a state

36. R6: *displays a value* precedes it requirement *made an ATP analysis*
   R16: *is done* is not describing a state

The paragraphs have also been analyzed. Paragraph 2 (sentences 4–10) and paragraph 3 (sentences 11–17) are both too long and violate R5.

The analysis of the text as a whole showed an average sentence length of 17.5 words, which violates R2.

D.3.3 Rewritten text

Automatic Sourcing Candidates

Usage

1 This window views all Sourcing Alternatives that the Automatic Sourcing Process used in its search for the best alternative. 2 These Sourcing Alternatives originate from the source set on the sourcing rule. 3 You can connect the Sourcing Rule to one of the following:
   - the Sales Part alone
   - the Customer and Sales Part
   - the Customer Address and Sales Part.

4 The candidates in this dialog box are the Sourcing Alternatives that were equally good. The process separated them through the Sourcing Objectives, i.e., the criteria. 5 The Selected check box is selected on the chosen alternatives.

Example

Five sourcing alternatives out of seven can deliver the whole sales quantity in time. 7 The dialog will show all five. 8 We assume that you only used Shipping Time as criterion. Two of the alternatives have the same total Shipping Time. 9 The process will select the Selected check box on these two. 10 The Automatic Sourcing Process will pick the first alternative of the two to supply the customer.

11 A Sourcing Rule consists of a Source Set and Sourcing Objectives. 12 The Source Set is a set of all Sourcing Alternatives. 13 Sourcing alternatives can be External Suppliers, Internal Suppliers, In-house Manufacturing,
The automatic sourcing will select the best alternative from this set. The best alternative will determine the Supply Code and, when appropriate, the Supplier on the Customer Order Line.

The Sourcing Objectives consist of a sequence of criteria. The automatic sourcing will use them when more than one sourcing alternative is suited to supply the customer order line. The automatic sourcing uses the following algorithm to select the best sourcing alternative:

1. From the Source Set, find all Sourcing Alternatives that can deliver the whole Sales Quantity in time.

2. If step 1 results in more than one Sourcing Alternative, use the criteria in the Sourcing Objective. This narrows down the possible alternatives.

3. If more than one sourcing alternative remains after step 2, pick the first alternative.

4. It is possible that step 1 resulted in no Sourcing Alternatives. Then, find the alternative from the Source Set that can deliver the whole order quantity earliest.

5. It is also possible that more than one Sourcing Alternative can deliver on the same earliest possible delivery date. Then, use the criteria in the Sourcing Objective to narrow down the possible alternatives.

6. If more than one sourcing alternative remains after step 5, pick the first alternative.

**Fields**

*Field descriptions without at least one full sentence are omitted*

**Supplier:** The Supplier selected by the Automatic Sourcing Process. This can be an Internal Supplier or an External Supplier. If this field is null, the site that takes the order will deliver it. That site is the site on the order line.

**Supply Site:** The supply site for this sourcing alternative. If this field is null and the Supplier field has a value, the Supplier is an External Supplier.
Selected: The check box that indicates whether this sourcing alternative is selected. All records that the Automated Sourcing considered as good alternatives during the final selection will have this check box selected. If several sourcing alternatives are selected, it means that these alternatives were equally good.

Ship Via Code: The default ship via code for this sourcing alternative. The Supply Code determines whether this Ship Via Code represents a direct or a transit delivery.

External Transport Lead Time: The external transport lead time when shipping parts. The External Transport Lead Time is expressed in calendar days, which includes weekends and holidays. This value is a key parameter when the system makes decisions during the sourcing process. It becomes a part of the total lead time when the system calculates the delivery dates for the orders.

Earliest Possible Delivery Date: The earliest possible delivery date, i.e. the first possible date when the end customer can receive the whole sales quantity from this sourcing alternative. This date is based on an Available To Promise (ATP) analysis or just lead time calculations. ATP analysis requires that the Availability or the Online Consumption Check option is selected. You can select this on the inventory part on the appropriate site.

ATP (Plannable Qty): The ATP at this sourcing alternative, i.e. this is the quantity that could be delivered in time in order to fulfill the end customer’s desired delivery date. This quantity can be less than the revised quantity. But then, the earliest possible delivery date will be later than the desired delivery date. The system can only analyze ATP if the Availability or the Online Consumption Check option is selected. You can select this on the inventory part on the appropriate site.

ATP Date (Available To Promise Date): The first date when full quantity can be picked from inventory to be delivered to the end customer. The system must first make an ATP analysis to be able to display a value in this field.

D.3.4 Analysis of the rewritten text

After rewriting the text such that all sentence and grammar rules are followed, the average number of words per sentence is 13.6, which obeys R2.
Appendix E

Test texts

In this appendix, we present the three test texts that we have used during interviews about rule applicability and effects. Each text exists in two versions. The two versions are made up of parts of the sample text taken at random from the original or the rewritten text. For each text, we here present what parts are from the original texts and which parts are from the rewritten texts; this information was not presented during our interviews.

E.1 Vertex Integration

This is a test text made from Sample Text 1. We have omitted one paragraph because of a technical error. The original text can be reconstructed if the numbered parts are taken from variant A and B according to 1:A, 2:B, 3:A, while the rewritten text naturally is 1:B, 2:A, 3:B.

E.1.1 Version A

1Vertex Inc. provides a third-party software solution for the U.S. and Canadian markets called Quantum for Sales and Use Tax, which calculates sales and use taxes. Taxes are calculated based on the state, county, city, and postal (zip) codes entered in customer order address fields. Through a subscription service, Vertex updates the taxation entities and percentages on a monthly basis. This greatly simplifies the process of correctly calculating taxes for sales. Vertex tax calculation is based on a nine-digit proprietary GeoCode. Each code represents a combination of taxation entities for state, county, and city. The first two digits identify the state, the next three identify the specific county, and the last four digits identify the city. This code is displayed in the Jurisdiction Code field in several IFS Applications win-
dows. The In City check box is used to indicate whether or not the address is within the city limits. This value will be used to determine whether or not to calculate city taxes.

2It is important to enter the correct address information for the tax calculation to work accurately with Vertex integration. Values for all the U.S. states are pre-installed in Enterprise/General Enterprise/State Codes. Use the List of Values whenever possible to select states for the state address fields. You must enter county, city, and postal (zip) codes manually.

Vertex uses generic tax codes to enable the retrieval of the appropriate tax amounts and percentages. You can establish these tax codes in Accounting/Rules/Tax Codes/Tax Codes. Then, you can select them in various windows via a List of Values.

Do not enter tax percentages if you have Vertex integration installed. The system automatically enters the percentages and calculates the tax amounts.

3Any time you make changes to any of the required address values and save the record, Vertex will use that new data to determine the GeoCode, which will be reflected in a changed value in the Jurisdiction Code field. If values in any of the mandatory address fields are missing or misspelled, you will receive an error message indicating that the address information is incomplete and that tax calculations cannot be performed. Valid address fields are needed for correct Geo Code retrieval.

E.1.2 Version B

1Vertex Inc. provides a third-party software solution for the U.S. and Canadian markets. This solution is called Quantum for Sales and Use Tax. It calculates sales and use taxes.

The tax calculation uses information about state, county, city, and postal codes from the customer order address fields. Through a subscription service, Vertex updates the taxation entities and percentages on a monthly basis. This makes it simpler to calculate taxes for sales correctly.

The Vertex tax calculation is based on a nine-digit proprietary GeoCode. Each code represents a combination of taxation entities for state, county, and city. The first two digits identify the state, and the next three digits identify the specific county. The last four digits identify the city. The Jurisdiction Code field displays this code in several IFS Applications windows.

The In City check box indicates whether the address is within the city limits. This value determines whether to calculate city taxes.

For tax calculation to work accurately with Vertex integration, it is important to enter the correct address information. Values for all the U.S. states are pre-installed in Enterprise/General Enterprise/State Codes. Use
the List of Values whenever possible to select states for the state address fields. County, city, and postal (zip) codes must be entered manually. Generic tax codes are used to enable Vertex to retrieve the appropriate tax amounts and percentages. These tax codes are established in Accounting Rules/Tax Codes/Tax Codes, and can be selected in various windows via a List of Values. Do not enter tax percentages if you have Vertex integration installed; the system will automatically enter the percentages and calculate the tax amounts.

When you change a required address value and save the record, Vertex uses that new data to determine the GeoCode. This changes the value in the Jurisdiction Code field. If any of the mandatory address fields is empty or has a misspelled value, you will receive an error message. This error message indicates that the address information is incomplete and that the system cannot calculate taxes. Vertex requires that the address information is valid in order to retrieve Geo Codes correctly.

E.2 Create Company In All Components

This is a test text made from Sample Text 2. The first part has been left out for future reference use. The original text can be reconstructed if the numbered parts are taken from variant A and B according to 1:B, 2:B, 3:A, 4:A, while the rewritten text naturally is 1:A, 2:A, 3:B, 4:B.

E.2.1 Version A

Prerequisites

- You must first define the Company Name and Company Type.
- You must first select Template or Company as Type of Source.

System Effects

This activity creates a company by these steps:

- Find all components that support the Create Company process. Also find out if the component supports company creation from template/company or if it is a component version from an old IFS Applications baseline that only supports the process using so-called default data that is independent of the selected source template or company. For more information, see the help file associated with the Component Registration window in IFS/Enterprise.
• All active Logical Units in all active components are accessed through a standard interface. This ensures that all data related to the Logical Unit is retrieved from the source (a company template or existing company) and then applied to the company that is being created or updated.

• Language translations are handled as follows:

  - When a new company is created, the translations selected from the source (company or template) are created in the new company for all active Logical Units.
  
  - When an existing company is updated, only the language translations that are used by that company will be updated. The updated language translations come from the source company.

"Note: The Update Company activity only adds basic data (and corresponding translations) to the current company.

• All errors during the process are logged.

• The process ends after having accessed all active Logical Units in all active components.

4It is important to point out that this activity will never fail, i.e., if errors are found, then these errors are logged and the activity will continue. This means that the company will be created at least in IFS/Enterprise.

It is also important to point out that updating a company always means that basic data and translations are added. Update Company will never remove or update any data in the current company.
E.2.2 Version B

1Prerequisites

- Company name and type of company must have been previously defined.
- Type of source, template or company, must have been previously defined.

System Effects

A company will be created as a result of this activity. The processing is performed according to the following:

- The system finds all components that support the Create Company process. In addition, the system examines each component to investigate if it supports company creation from Template or Company. Components from old IFS Application baselines only support default data, which is independent of the selected Type of Source. For more information, see the help file for the Component Registration window in IFS/Enterprise.

- The system accesses all active Logical Units in all active components through a standard interface. This ensures that the system retrieves all data for the Logical Unit from the selected source template or company. The system then applies this data to the company that it creates or updates.

- The system ends the process.

The system handles language translations as follows:

When you create a company

The system uses the translations that you selected to create new translations in the company for all active Logical Units.

When you update an existing company

The system only updates the language translations that the company uses. The updated language translations come from the source company.
Note: The Update Company Activity only adds basic data (and its translations) to the current company.

The system logs all errors during the process.

It is important to point out that this activity never fails. If the system finds errors, it logs them and the activity continues. This means that the process creates the company at least in IFS/Enterprise.

It is also important to point out what happens when the system updates the company. This always means that the system adds basic data and translations to the company. Update Company never removes or updates any data in the current company.

E.3 Automatic Sourcing Candidates

This is a test text made from the definitions in Sample Text 3. One definition has been left out for future reference use and one did not need any rewriting and is therefore also omitted. The original text can be reconstructed if the numbered parts are taken from variant A and B according to 1:A, 2:A, 3:A, 4:B, 5:B, 6:A while the rewritten text naturally is 1:B, 2:B, 3:B, 4:A, 5:A, 6:B.

E.3.1 Version A

1Supplier: The supplier selected by the automatic sourcing process. This can be an internal supplier or an external supplier. If this field is null, the order will be delivered from the site taking the order, i.e. the site on the order line.

2Ship Via Code: The default ship via code for this sourcing alternative. Depending on the supply code, this ship via code represents a direct or a transit delivery.

3External Transport Lead Time: The external transport lead time when shipping parts. The external transport lead time is expressed in calendar days, which includes weekends and holidays. This value is a key parameter for decision making during the sourcing process, and becomes a part of the total lead time when the delivery dates for the orders are calculated.

4Earliest Possible Delivery Date: The earliest possible delivery date, i.e. the first possible date when the end customer can receive the whole sales quantity from this sourcing alternative. This date is based on
an Available To Promise (ATP) analysis or just lead time calculations. ATP analysis requires that the Availability or the Online Consumption Check option is selected. You can select this on the inventory part on the appropriate site.

5**ATP (Plannable Qty):** The ATP at this sourcing alternative, i.e. this is the quantity that could be delivered in time in order to fulfill the end customer’s desired delivery date. This quantity can be less than the revised quantity. But then, the earliest possible delivery date will be later than the desired delivery date. The system can only analyze ATP if the Availability or the Online Consumption Check option is selected. You can select this on the inventory part on the appropriate site.

6**ATP (Available To Promise Date):** The first date when full quantity can be picked from inventory to be delivered to the end customer. This field only displays a value if an ATP analysis is done.

**E.3.2 Version B**

1**Supplier:** The Supplier selected by the Automatic Sourcing Process. This can be an Internal Supplier or an External Supplier. If this field is null, the site that takes the order will deliver it. That site is the site on the order line.

2**Ship Via Code:** The default ship via code for this sourcing alternative. The Supply Code determines whether this Ship Via Code represents a direct or a transit delivery.

3**External Transport Lead Time:** The external transport lead time when shipping parts. The External Transport Lead Time is expressed in calendar days, which includes weekends and holidays. This value is a key parameter when the system makes decisions during the sourcing process. It becomes a part of the total lead time when the system calculates the delivery dates for the orders.

4**Earliest Possible Delivery Date:** The earliest possible delivery date, i.e. the first possible date when the end customer can receive the whole sales quantity from this sourcing alternative. This date is based on an Available To Promise (ATP) analysis only if the inventory part, on the appropriate site, has the availability or the online consumption check option selected. Consequently, this means that in other cases this date is only based on lead time calculations.
5 **ATP (Plannable Qty):** The ATP at this sourcing alternative, i.e. this is the quantity that could be delivered in time in order to fulfill the end customer’s desired delivery date. If this quantity is less than the revised quantity, the earliest possible delivery date will be later than the desired delivery date. ATP is only analyzed if the inventory part, on the appropriate site, has the availability or the online consumption check option selected.

6 **ATP Date (Available To Promise Date):** The first date when full quantity can be picked from inventory to be delivered to the end customer. The system must first make an ATP analysis to be able to display a value in this field.
Index

AECMA Simplified English, 79
anaphor, 33
anaphora, 33
anaphora resolution, 33
antecedent, 33
approved words, 110
articles, 115
black box view, 40
Bleu, 41
Caterpillar Fundamental English, 77
Caterpillar Technical English, 78
cloze procedure, 23
cohesion, 17
cohesive tie, 17
compound terms, 114
compounds, 31
comprehensibility depth, 13
comprehension, 12
concept, 66
conformance problem, 82
conjunction, 19
connotation, 32
contractions, 113
controlled languages, 57
convergence, 92
definition, 66
denotation, 32
EasyEnglish, 81
ellipsis, 18
Ericsson English, 80
expressive function, 53
extended vocabulary CL, 78
hyperonymy, 73
hyponymy, 73
IFS Style and Writing Guide, 98
ILSAM, 78
informative function, 53
interactive authoring systems, 86
lexical ambiguity, 30
lexical cohesion, 19
limited vocabulary CL, 78
machine translation, 34
Microsoft Manual of Style for Technical Publication, 46
monolingual CL, 78
mood, 52, 117
multilingual CL, 78
Nist measure, 43
noun attachment, 31
noun clusters, 114
onomasiological approach, 70
operating instructions, 49
PACE, 78
PEP, 78
person, 118
phrasal verbs, 121
precision, 91
prepositional phrase attachment, 31
prescriptive approach, 86
procedures, 50
pronominal anaphora, 33
proscriptive approach, 86
pseudo-standards, 67
psycholinguistic redundancy, 17
pure standards, 67

readability, 13
readability index, 23
reading goal, 13
recall, 91
red light/green light systems, 86
reference, 18

Scania Swedish, 81
semantic analysis, 31
semantic role, 19
semasiological approach, 69
sequences, 51
structural ambiguity, 31
style for technical writing, 49
style guides, 45
sublanguages, 58
substitution, 18
syntactic ambiguity, 31
syntactic analysis, 30
syntactic cues, 53

target language equivalent, 72
tense, 52
tenses, 116
term, 66
terminology, 65
terminology management, 65
text functions, 53
The Chicago Manual of Style, 46
thematic roles, 19
translatability, 27
translation memory, 90
two-word verbs, 121

usefulness, 92
verb attachment, 31
verb forms, 116
verbals, 118
vocabulary, 66
vocative function, 53
voice, 51, 117

warnings, 51
word clusters, 114
writer certification system, 83