Intelligent Computer-Assisted Instruction in Syntactic Style

Julie Payette

Technical Report CSRI-247
October 1990

Computer Systems Research Institute
University of Toronto
Toronto, Canada
M5S 1A4

The Computer Systems Research Institute (CSRI) is an interdisciplinary group formed to conduct research and development relevant to computer systems and their application. It is an Institute within the Faculty of Applied Science and Engineering, and the Faculty of Arts and Science, at the University of Toronto, and is supported in part by the Natural Sciences and Engineering Research Council of Canada.

Intelligent Computer-Assisted Instruction in Syntactic Style

by

Julie Payette

A Thesis submitted in conformity with the requirements
for the Degree of Master of Applied Science in the
Department of Electrical Engineering,
University of Toronto

October, 1990

It is important to understand [Harold Somers's] idiosyncratic conception of science (theory) vs. engineering (applications) because he is discussing the relationship between case grammar, a type of linguistic science, and computational natural language processing, which, for all the rhetoric about artificial intelligence and modeling of human cognitive processes, is essentially engineering.

Stanley Starosta in his review of H.L. Somers
"Valency and Case in Computational Linguistics."
Edinburgh University Press, 1987.
In Machine Translation 5(1), March 1990, p. 81.

Abstract

Writing with style is a skill that can be perfected through practice and experience. Yet before a writer's style is allowed to flourish and flow freely, the writer must be taught the principles and conventions of the language, beyond those of grammar, that contribute to stylistic effectiveness.

This thesis describes a computer-assisted language instruction system that is meant to assist language instructors in teaching principles of syntactic style to students of English. Based on artificial intelligence techniques in natural language processing and specifically developed for stylistic instruction, the system, named STASEL for Stylistic Treatment at the SEntence Level, analyzes free-form input sentences interactively. STASEL's stylistic processing capabilities reside in two separate modules: a syntactic style analyzer (SSA) module that detects stylistic problems in diction, usage, and sentence structure, and a goal-directed style analyzer (GSA) module that determines the structural clarity of the input. The system summarizes its findings in a structured output message that provides instructional feedback.

STASEL is implemented in PROLOG and several ouput examples generated by the system have been included in the thesis.

An important contribution of STASEL is its ability to provide stylistic guidance according to the specific writing goals of clarity and conciseness. In an attempt to remedy some of the deficiencies of existing instructional software, STASEL's design lays the groundwork for the creation of intelligent tutoring systems for teaching writing.

Acknowledgments

I would like to extend my first and very sincere acknowledgments to my supervisors, Graeme Hirst for theoretical and writing guidance, and Michael Stumm for psychological support, verbosity control and common sense perspective. They were times when I saw no end to this work, but they made me see the light.

I am also very grateful to Chrysanne DiMarco who spent time to provide me with much appreciated research guidance during one of the most tense moments of this thesis.

Many thanks and best wishes to the members of the EECG and NLU groups for their presence, conversation, and encouragement. Un merci tout spécial à Orran pour son temps, son appartement, et son soutien constant.

Finally, I wish to acknowledge the support of my friends, in Toronto and in Montreal, whom I have terribly neglected in the last few months, but who have all had to listen patiently, at some point or another, to lengthy explanations and endless justifications of my work. At the forefront of this group comes Mathew. Merci!

Financial support for this thesis was provided by NSERC.

Enfin, je tiens à remercier ma famille qui, malgré la distance, a su m'appuyer et me soutenir sans relâche. Sa présence me manque beaucoup. Je dédis ce travail à mon père, André, qui m'a transmis le goût de l'effort, à ma soeur, Maude, qui m'est si chère et à qui je souhaite tous les succès du monde, et plus particulièrement à ma mère, Jacqueline, qui m'a inspiré le sens du dépassement de soi et à qui je dois tant.

Et je n'oublie pas, maman, *qu'il y a toujours place pour amélioration ...*

Contents

List of Figures

List of Examples

Chapter 1

Introduction

On n'arrive au style qu'avec un labeur atroce.[1]

It is a widely accepted view that learning to write a language well is a long and arduous process. It is also well known that teaching the techniques of writing requires patience, insight, and a solid background in the structures and rules of the language being taught. Much emphasis is usually given, in language classrooms, to the teaching of grammar, but we sometimes overlook the importance of style and its effect on the resulting composition. Yet, style remains a fundamental characteristic of good writing: it brings elegance and power to prose, transcending the written forms and highlighting the meaning; it is the ultimate vehicle, beyond grammar and content, through which a writer truly communicates with the reader.

However, if we believe Flaubert whose famous quotation opens this work, the task of mastering style does not come easily. This is especially true when viewed from the perspective of the language instructor who is faced with the challenge of *teaching* style. Stylistic instruction may be challenging, but it is also rewarding. Corbett [Cor86] evokes the task with the following positive words:

> I have found that students are invariably fascinated by style—not only because it offers something new and different but also because it provides an element of fun in changing words and shifting parts. (p.25)

Typically, teaching style in language courses has two objectives: to teach how to analyze and recognize someone else's style and how to improve one's own. Yet before they can develop a style of their own, students must be taught the rudiments of stylistic effectiveness[2]. In other words, they must learn the principles and conventions of the English language, beyond those of grammar, that contribute to achieving good style in formal writing. Stylistic rudiments, in themselves, do not ensure that the resulting prose will be coherent and effective, but they represent an essential step in the acquisition of basic composition skills.

[1] A quotation from Gustave Flaubert, French novelist of the 19th century, which literally translates as *One reaches style only through excruciating labour*

[2] Effectiveness and correctness are sometimes confused. The first means that a piece of writing achieves its purpose; the second that it is written according to the rules of grammar, usage, spelling and punctuation. Style is concerned with effectiveness, once the conventions that determine correctness have been respected.

1.1 Thesis Description

This thesis describes a tutorial approach to computer-assisted language instruction (CALI) that is meant to assist language instructors in teaching principles of syntactic style. The objective of the research is to apply the techniques of artificial intelligence (AI) to an aspect of language instruction, stylistic processing, which has so far received little attention in the field of CALI. This research has led to the development of a prototype natural language system that processes sentences and provides stylistic feedback to the student.

The system, named STASEL for *Stylistic Treatment At the SEntence Level*, is a sentence-level tutor program which aims at (1) teaching stylistic rudiments and (2) providing guidance as to how English structure influences stylistic shape. Though it can perform some of the functions of a conventional "style checker", flagging diction and usage errors in written texts, STASEL goes beyond such systems. It departs significantly from other experimental or commercial stylistic processing programs in that it displays all of the following features:

1. it is specifically designed for educational purposes and it emphasizes pedagogically sound teaching strategies;
2. it is based upon a comprehensive parser that can recognize stylistic features and analyze an important range of English structures;
3. it generates an explicit representation of a sentence's syntactic structure that is specifically suited for stylistic processing and for teaching the mechanics of English sentence construction;
4. its design is highly modular, thus facilitating the extension and refinement of implemented features, or the addition of new functions.

STASEL's most important contribution to the field of computer-assisted language instruction lies in its ability to provide stylistic guidance according to specific writing goals. The system not only detects conventional points of style such as usage, wordiness, and positional problems, but also explicitly identifies the building elements of the input sentence. The analysis of the inner structure allows the system to comment on the stylistic shape of the sentence according to precise stylistic goals. For the purpose of demonstration, STASEL analyzes sentences according to the goals of clarity and conciseness, and provides guidance regarding two categories of syntactic stylistic variants[3]: diction and sentence construction.

STASEL is an interactive system which accepts a student's composition, a sentence at a time, and responds with a set of stylistic messages, a diagnosis of clarity, and a description of the sentence structure. Example 1.1 shows a typical STASEL response for a clear and stylistically correct sentence, while Example 1.2 demonstrates how the system responds to structurally awkward and stylistically poor inputs.

STASEL provides instruction for intermediate students of English who have already mastered complex syntactic patterns, but the system only processes input free of grammatical and spelling errors.[4] Although the tutorial can be applied in an English composition class, its emphasis on structural elements makes it particularly suitable for second language teaching, where the subtleties in the arrangement of sentence components must be taught

[3] See section 2.1 for a more detailed discussion of stylistic variants.

[4] In its present state, STASEL does not include grammatical and spelling corrector modules, but its modular design ensures that the incorporation of such a module is readily feasible.

```
------------------------------------------------------------------

ANALYSIS OF:  John, my brother who is tall, is an engineer.

------------------------------------------------------------------

 <=> GOOD: stylistically correct

------------------------------------------------------------------

 <=> SENTENCE IS CLEAR: positive interruption

------------------------------------------------------------------

 <=> Structure: simple clause with interruption

    interruption -->
        apposition: np with relative clause

    clause -->
        subj: simple noun phrase (np)
        comp: simple noun phrase (np)

------------------------------------------------------------------
```

Example 1.1

more rigorously. STASEL has the capability of analyzing lexical, structural, and positional characteristics of a sentence, but it is not equipped to process its semantic content. Thus, it cannot comment on the meaningfulness of the input sentence. The modularity of the system's design will allow semantic processing to be incorporated as an additional function, as suitable techniques are developed.

STASEL's design is intended to lay the groundwork for the creation of intelligent tutoring systems (ITS) for teaching writing. The development of such expertise is an attempt to remedy some of the deficiencies and limitations of existing CALI technology. To date, CALI programs have made only limited use of AI techniques, and for relatively simple language tasks. This research is premised on the notion that AI techniques will enable programs to tackle language instruction more "intelligently", so that in the future, programs will be able to communicate meaningfully with the students, to detect and correct errors in writing, and to provide students with appropriate, individualized instruction based on their past performance. The resulting intelligent language courseware will provide effective and interactive guidance to the student, thereby relieving the instructor of repetitive remedial tasks.[5] The intent of ITS research is not to replace the language instructor altogether. A program like STASEL is an instructional tool that can interact successfully with a student, but that still necessitates the direction and supervision of an instructor. As Neuwirth [Neu89] points out, it is unrealistic, if not pretentious, to expect that intelligent tutoring systems will take over the language classroom, for

> no existing intelligent tutoring system exhibits the entire range of abilities we

[5] In speculating about intelligent systems, one must realize that current implementations incorporate *experimental* AI techniques, and thus are unlikely to be seen in routine educational applications for some years to come.

```
---------------------------------------------------------------------

ANALYSIS OF:  Learning to speak a second language in a place
              where the language spoken is different can be
              difficult and writing it is almost impossible.

---------------------------------------------------------------------

First Clause:
--------------

  *  1 passive detected

Second Clause:
---------------

 [W] wordy construct with "impossible" and verb "to be".
     It makes prose insipid. If possible, discard it altogether


---------------------------------------------------------------------

 *** sentence is unclear ***

  >> complex structure in the subject of the first sentence
  >> unbalanced compound structure

---------------------------------------------------------------------

 <=> STRUCTURE: compound of sentences

    first sentence:
        clause -->
            subj: complex >> gerundival with complex complement
            comp: adjectival phrase

    second sentence:
        clause -->
            subj: gerundival clause
            comp: adjectival phrase

---------------------------------------------------------------------
```

Example 1.2

> usually associate with the best human tutors. Despite the word "intelligent" in the name, and a *goal* of high performance, existing intelligent tutoring systems are more modestly and commonly viewed as explorations into how computer technology can be used to specify, or more precisely, to formalize theories of tutoring. (p.46)

One cannot help feeling awed at the task of constructing a truly intelligent computer program for language instruction, but the means of achieving this goal are more clearly defined due to advancements in intelligent CALI (ICALI) research. ICALI is a multidisciplinary field, which is not only concerned with pedagogy, but also with the fundamental properties of language. Building an *intelligent* tutoring system involves a large number of tasks and combines the expertise of the following fields:

Education: First and second language educators elaborate the teaching methodologies and provide the pedagogical assumptions underlying the system's design.

Psychology: Cognitive psychologists describe the learning process and provide a theoretical framework that can be used to tailor the system's response to the learner's needs and aptitudes.

Linguistics: Linguists define the formalisms necessary to represent the internal properties of language.

Computer Science and Engineering: Computer specialists working in AI offer techniques both for the implementation of intelligent systems and for the computational processing of the linguistic knowledge.

When focusing on the development of systems that intend to replicate, to some extent, the teaching behaviour of human language instructors, it is important to emphasize the potential benefits ensuing from advancements in artificial intelligence, and particularly, in natural language processing (NLP) research. In fact, many ICALI researchers believe that the incorporation of AI and NLP strategies has the potential to radically improve the efficiency and communicative ability of current and future instructional programs ([Las89, Und89, BL89, Cha89, Neu89, Far89a, Mul89, Ham90, Fer89] and many more).

The remainder of this chapter presents an overview of the field of ICALI and attempts to position the contribution of STASEL within the larger framework of intelligent tutoring system research. Chapter 2 focuses on the more specific issue of stylistic instruction, and outlines the goals and significance of research in this field, detailing the theoretical basis for the design of STASEL. This chapter also describes current computer systems that have implemented some form of stylistic processing. The next three chapters describe the implementation of STASEL and provide examples of its operation. Two of these chapters (Chapters 4 and 5) concentrate on the stylistic processing capability of the system. Chapter 6 discusses the strengths and limitations of the system while laying the foundation for future research. A glossary of grammatical and stylistic terms has been provided in appendix A for those readers who are not familiar with linguistic jargon, and in particular, with the stylistic vocabulary used in this thesis. Several output examples demonstrating the operation of STASEL are presented in Appendix B.

1.2 Research in Intelligent CALI

Intelligent computer-assisted language instruction is a sub-field of artificial intelligence which draws expertise from computer science, linguistics, language pedagogy and cognitive psychology. Its theoretical motivation is to model the linguistic knowledge and abilities of the human mind, and investigate the highly complex process that constitutes the psychology of language teaching and learning. It is also concerned with building practical applications in its attempt to create computerized emulations of intelligent language-related behaviours. In contrast with earlier CALI efforts, which mostly produced automated drill-and-practice exercises deprived of flexibility and communicative ability, *intelligent* CALI has based the development of efficient instructional software on the use of AI techniques in knowledge representation and natural language processing.

Though still in its infancy, the potential of ICALI research is widely recognized in the literature and is attributable, not only to the increased degree of flexibility and control it can offer the student,

> AI tools may help us overcome some of the major drawbacks in CALI software. Programs could be more interactive, departing from the rigid objective questions, and allowing freer communication between the student and the computer. [Far89a, p.236]

but also, to the potential benefits ICALI can bring to language instruction as a whole:

> As ICALI software becomes more and more able to assume some of the onerous aspects of language teaching, language teachers can view their roles in different ways and begin to look for alternatives to the traditional classroom. [BL89, p.4]

> ICALI systems emphasize a design process centered on creating a rich, responsive learning environment which will encourage greater cognitive exploration and, capitalizing on student interest, foster increased positive learning attitudes. [Duc88, p.23]

Despite almost two decades of research and development in computer-assisted instruction, most current CALI programs still present deficiencies in communicative ability and user flexibility. In my opinion, the seemingly slow evolution and lack of breakthroughs observed in the field, at least in the first decade, arose from the incapability of language educators and computer scientists to work together and collaborate in the elaboration of efficient and adequate software.[6]

1.2.1 First and Second Generation CALI

Computer specialists are responsible for the development of first-generation CALI systems. Early systems were characterized by carefully programmed, deterministic sequences of questions that would only accept a very restricted set of answers. In many cases, these systems would reject a student's answer simply because it was not listed in the program among the acceptable ones. Developed mostly by programmers who knew very little about the principles and techniques of teaching, early CALI systems were very adept at drill-and-practice,

[6] The unwillingness of educators and computer scientists to appreciate each other's contribution in CALI research can be observed throughout the literature. For a particularly harsh denunciation of the potential of AI techniques in CALI, from the point of view of an education specialist, see Ng [Ng84].

often implementing the exercise with many "colours, accents and other surface attributes" [Las89, p.8], but showing little teaching capability other than printing questions and correcting answers. More importantly, such systems did not let the student use the language freely. An underlying theme of criticism in early CALI is that it seemingly had taken a wrong direction in two important ways [Und89]:

1. Computer activities were based on questionable pedagogical principles (if any).
2. CALI tended to treat the computer as a one-way system, a drillmaster which spent more time telling students what they should know than it did encouraging them to discover things for themselves.

It was argued, then, that CALI's instructional quality could not improve without the necessary inclusion of language teaching methodologies:[7]

> [It is] vital that language teachers overcome their natural scepticism towards technical aids and seriously consider the implications of the computer's role in language learning and teaching. Only by becoming involved at an early stage will they be able to determine the course of events in this important sphere of educational technology and ensure that the right kind of courseware is created. Failure to do so will result in amateurs who know little about language teaching but a lot about computers taking the initiative and producing language learning and teaching programs of inferior quality. The danger signs are already there. [Dav82, pp.vii–viii]

In response to this need, a second generation of CALI programs saw the emergence of pedagogical awareness and the need to involve teaching specialists in the design of CALI programs.

> Although computer scientists can be and often are the chief designers of ICALI systems, they are often not trained to deal with the communication, linguistic and educational factors which must be considered in developing ICALI systems. [...] The skills and insight of language teachers are crucial in most areas of an ICALI project. Teachers who can put into words and formalize their many years of experience are probably the most valuable people for designing expert and student models and pedagogical strategies. [BL89, pp.9–1]

As a result of serious involvement of language educators, programs became more pedagogically oriented and attempts were made to base system development on a more communicative approach to language learning. A notable example is the tutorial approach introduced by Weischedel, Voge, and James [WVJ78], who built a prototype system, based on natural language techniques, to evaluate the reading comprehension of second language learners of German. Developed in the late 1970's, the system is particularly interesting because it implements several pedagogical and computational ideas that depart from earlier approaches. The system presents students with a text in German and then asks questions designed to test the student's comprehension of the text. The student practices the language by typing sentences as answers. The intelligent tutor searches for possible problems in the student's response, including errors in syntax, semantics, understanding, and

[7] The need to introduce practical pedagogical techniques in the design of CALI programs is still widely recognized among researchers ([Bur88, Und89, Bur90, Yaz89, Fer89]).

spelling. Computationally, the interest of the prototype lies in the fact that the processing is modular: a syntactic component, capable of handling ill-formed sentences, first analyzes the input, then passes a complete parse on to the semantic component, which judges the validity of the answer. Pedagogically, the system was one of the first to imitate communicative classroom practices by providing an interactive, free input[8] tutorial environment for the student. Weischedel, Voge, and James's design has become, in some sense, the norm in subsequent CALI developments, first, because it introduced a rigorously sequential organization of the syntactic and semantic modules, thereby minimizing the amount of data needed at each stage of the program, and second, because it laid the groundwork for communicative software and intelligent tutoring system design (described in Section 1.3.2).

1.2.2 Toward Intelligent Language Processing

Despite the contribution of Weischedel et al., it took another decade before research in computer-assisted instruction reached its present state and became known as "intelligent", or "third-generation", CALI. Research in the second period can be regarded as a reflection aimed at formally defining the necessary features of future CALI programs. The trend had already been set toward more pedagogically oriented and communicative software, but the field had yet to realize the necessity to process language more conclusively, more explicitly, and more intelligently. One of the strongest influences on modern language teaching and second-generation CALI, the communicative school[9], can be used to illustrate this reflection.

Still a prevalent theory in some current CALI system development (see [KA88]), the communicative school sees language learning as the acquisition of the ability to communicate, rather than the mastery of particular grammar structures. This approach places emphasis on the function of language as a communication medium that is more concerned with meaning than structure.

> Almost from its conception, CALI has had to confront a cruel disparity between its capabilities and the demands of modern pedagogical theory. CALI developers were exploring a new application of the written word despite the nearly universal precedence that learning theory accords the spoken medium; and while program designers were learning how to make smoothly interactive drills and increasingly accurate morphological generators and analyzers, language teachers were learning to distrust drill entirely, to de-emphasize morphology, and to focus on communicative methods as the surest routes to language acquisition. [Mul89, p.32]

The precepts of the communicative school found their way into the classroom, embodied by the very popular practice of language laboratories[10], and into the design of CALI pro-

[8] In WVJ's design, input flexibility is obtained by the use of a parser in the syntactic component, but the system is severely limited by the need to build a detailed knowledge representation in the semantic component for each text and each set of questions. This limitation increased the complexity of the program to such an extent that only a few dialogues (with limited grammar and vocabulary) could be implemented.

[9] Communicative competence plays a central role at all levels of language learning. Its importance in today's language teaching practices is expressed by the proliferation of teaching methods that aim at achieving fluency: language laboratories, conversation classes, immersion schools. [Las89], [Jon86]

[10] Language laboratories aim at developing communication skills in the language being learned by making students orally repeat, correct, or complete utterances given to them in a recorded lesson. A teacher occasionally listens and corrects the student's pronunciation and mistakes, but most often, the student is left practicing on his own. [Har83]

grams. Gwyn Jones [Jon86], for example, implemented a system which can hold a simple conversation with a student about dietary questions. The system uses a form of pattern matching to break down the input sentence and responds with dietary advice for the topic of the query. In all, the system is unsophisticated, limited in its input, and incapable of diagnosing errors let alone correcting them.

Gwyn Jones's system is based on an important but slightly paradoxical premise: that language instruction must concentrate on communicative ability rather than (or, one could almost say, to the detriment of) structure. Though it is indisputable that the development of highly communicative programs greatly improves the interactive efficiency of computer-assisted instruction, and thus, that the development of "natural" student/machine means of communication should be included as a design goal of most CALI systems, it is unrealistic to pretend to emphasize meaning while overlooking the importance of the language structure. In other words, besides communicative skills, CALI programs must display an explicit knowledge of the structures of the language they intend to teach.

This need for the incorporation of formal linguistic knowledge becomes even stronger when viewed from the perspective of available technology. Considering that voice recognition technology is not yet available in classrooms and that the current material displays only writing and reading (and sometimes listening[11]) capabilities, researchers in CALI have little choice but to place emphasis on the written form of the language of which syntax is an essential, if not the primary, building block.

From the above discussion of past CALI research, one can conclude that sophisticated interactive systems which display linguistic as well as pedagogical knowledge are needed to give more freedom in the student's use of language. It is also clear that researchers must investigate new instructional vehicles to carry the goals and ambitions of the field, as previous ones have failed to provide adequate flexibility. As we have seen, the assessment of computerized drill and practice exercises, which still continues to account for a substantial portion of present CALI production, has been disappointing, primarily because these exercises know little about grammar, are excessively deterministic, and show no pedagogical insight.

Thus, ICALI research entered its third generation when it decided to move beyond deterministic "dumb" exercises or simple conversation-oriented tutors. It now undertakes the creation of instructional software with the lessons of earlier efforts in mind: present-generation CALI programs aim at establishing a *natural* communication with the student, at organizing teaching sessions according to sound pedagogical principles, and at processing student input according to a formal representation of the language under study. Underwood describes the latter feature as paramount to the evolution of ICALI:

> As we enter the second decade of using computers for language instruction, we realize that our efforts were not necessarily the best way to proceed, and we are now actively seeking alternatives. [...] It seems that the real breakthrough in the CALI of the 1990's will not come primarily from new machines or from new media, but from a more complete and more sophisticated handling of the language. *Intelligent Language Processing* is the essential ingredient which may ultimately make all CALI worthwhile. [Und89, pp.82–83]

[11] Some work has been done in ICALI to develop an alternative educational environment using non-text media. The environment under study is called *hypermedia*. Its objective is to link video, audio, graphic, and animation technology to the instructional process so that student can freely explore vast quantities of information. See Underwood [Und89, pp.77–80] for a more detailed description of hypermedia.

It is fundamental to this research that the strategies and principles of natural language processing (NLP) can make intelligent language processing a reality in a language teaching environment. The system developed in this thesis describes how such techniques are applied within the framework of an intelligent tutoring system, keeping in mind the concerns uncovered in earlier research.

1.3 Types of ICALI Systems

Existing CALI programs come in various forms: they are either instructional or for testing the student; they come in the form of drill and practice exercises or authoring packages[12]; they are text-based or make use of graphic facilities; some deal with vocabulary acquisition, conjugation of verbs, reading comprehension or spelling, while the most ambitious ones move beyond simple phrases to sentences or even to discourse considerations.

The vast majority of available teaching programs, referred to from here as conventional CALI programs, present a number of important deficiencies. In general, only a limited approximation of the knowledge a teacher possesses about teaching a subject is incorporated into these programs. The questions and answers are determined in advance ("canned"), and the program leaves little control and flexibility to the learner. By storing fragments of *performance* rather than modelling aspects of *competence*, conventional programs rely on accuracy of response, and thus, are commonly unable to accept partial answers or accommodate the different levels of ability and creativity found in students. In all, the degree of real interaction with the learner is highly restricted.

The work underway in the field of ICALI, while trying to provide an answer to present inadequacies, aspires beyond the conventional approach of drill and practice programming. In order to create interactive computerized courseware, researchers in artificial intelligence and computer-assisted instruction have developed new platforms on which intelligent CALI design can be based:

1. Dialogue systems
2. Intelligent tutoring systems
3. Microworld
4. Open-ended text processing

The following sections present each of these platforms in turn. A special emphasis has been given to intelligent tutoring systems over the other types, because of its closer relationship to the design of STASEL.

1.3.1 Dialogue Systems

The objective of a dialogue system is to enable a computer to simulate a plausible conversation with the student. Its function, much like that of an expert system, is to give advice and help students by responding to queries about the subject being taught. To be useful, a dialogue system must be able to deal with a range of subjects related to an open-ended main topic. For example, in language instruction, the dialogue must contain explanations about

[12] Authoring packages are template programs in which the form of the exercises is fixed but which allow teachers to insert data and create their own examples.

the rules and conventions of the language under study, and how these rules and conventions are applied. Linguistic problems encountered during the interchange are usually resolved, but not necessarily corrected, by means of system-initiated queries known as *clarification dialogues* [13].

Because they respond to student-initiated queries, dialogue systems are not very suitable for teaching applications in which the goal is to impart precise knowledge to the student. Moreover, linguistic and computational problems, such as the resolution of ambiguity in the input or the implementation of a comprehensive language knowledge base, still need to be solved before such a system can become practical. These considerations may account for the little attention systems of this sort have received in ICALI, although they are receiving much attention in other areas.[14]

1.3.2 Intelligent Tutoring Systems

An intelligent tutoring system is a specialized type of expert system which attempts to emulate the role of an experienced tutor. The objective of intelligent tutor programs is to provide *personalized* instruction by presenting controlled exercises geared to the student's ability level and by responding with individualized corrective feedback. In language instruction, the material presented can be aspects of grammar, style, and rhetoric of the language being taught.

To be practical, an ITS for language instruction must be capable of understanding the language under study, and of modelling student behaviour in order to adapt its approach accordingly. This can involve not only employing *expert*, *student*, and *pedagogical* models, but also incorporating enough linguistic skills to interact with the student in the target language. Such considerations have resulted in the formulation of an ITS architecture for language instruction (see Figure 1.1). The standard view, outlined in the works of Farghaly [Far89a], Neuwirth [Neu89], Duchastel [Duc88] and Ferney [Fer89] (among others), identify four principal components of the architecture:

1. Linguistic component
2. Pedagogical component
3. Expert module
4. Student model

The *Linguistic component* is a natural language communication interface that understands the student input and generates system responses. Using a lexicon[15] and a parser[16], the linguistic component analyzes the input, checking whether it is well formed (syntactic parse) and, whether it makes sense (semantic parse). The parser of the linguistic component draws its expertise from the *expert module*, which contains an explicit representation of the language under study. To facilitate the operation of the parser, the linguistic component and the expert module are often designed as a single entity.

[13] The term *clarification dialogue* comes from Winograd [Win84].

[14] "In industry, dialogue programs serve as front-end interpreters for expert systems, in fields ranging from medicine to the analysis of computer hardware problems." [Und89, p.77]

[15] A lexicon is a dictionary in which the words understood by the system are listed along with their grammatical and, possibly, semantic attributes.

[16] A parser is a program that implements the grammar rules and structures of a target language.

The result of the linguistic analysis is passed on to the *pedagogical component*, which decides how to respond, in consultation with the expert module and the *student model.* Once it has decided upon the most suitable course of action, the pedagogical component sends its conclusions to the linguistic component, which generates the system output to the student.

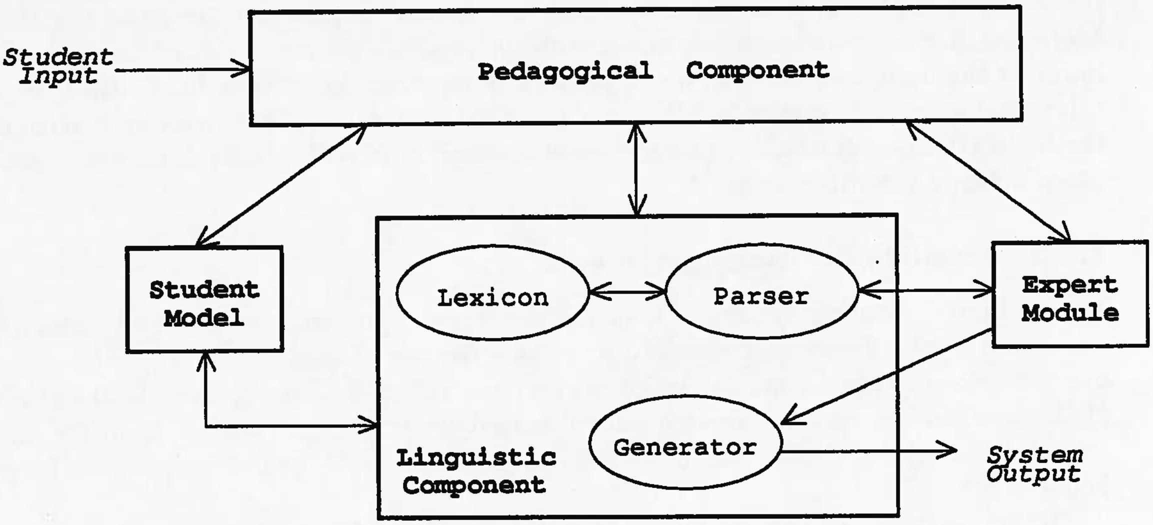


Figure 1.1 An architecture for an ITS

The ITS architecture exhibits the following noteworthy features:

- The pedagogical component is the most prominent component of the architecture. Its main task is to execute the tutorial strategy and make decisions about the teaching sequence according to

  - the pedagogical principles that have been coded into it;
  - information about the student provided by the student model.

  The pedagogical component usually contains a variety of techniques for dealing with syntactic or semantic errors picked up by the expert module. It might also contain procedural rules that indicate whether or not to intervene in a given situation. For example, if a parse fails, the pedagogical component determines which rules have been violated, classifies them, and decides on the appropriate response. Ideally, the pedagogical component should be able to handle ill-formed inputs, but more importantly, if it is to be able to teach anything, it must allow mistakes to be made precisely in the aspect of language upon which the student is instructed.

- The linguistic component should have the capability of understanding the meaning of the input. Yet since semantic analysis is trickier to perform than syntactic analysis, implementations of this component usually treat the input semantically only if the application requires an interpretation of the input (such as in reading comprehension). Semantic analysis may be necessary, though, in all cases, if ambiguities in the language used make it impossible for the parser to decide on the grammatical structure of a sentence without semantic information.

- The student model enables the system to tailor its response and individualize its instruction to the needs and ability level of each student. Such a model must at least be able to account for what the student knows (history and performance evaluation), and for what he needs or wants to know (teaching objectives). This makes the design of student models particularly problematic, since it is difficult to deduce from a student's observable performance precisely what his goals and aptitudes are. Research pertaining to the development of student models, also called *user modelling*, has generated much interest in computer-assisted instruction in general ([KW88, Car85, Sel87, CJ86, KF88] among others), but to date, only a few programs have implemented some form of student model.[17]

Designing intelligent tutoring systems is an difficult and exacting task: it requires the formulation of explicit and comprehensive knowledge about the topic, the user and the teaching methodology; it consumes vast amounts of computational power and necessitates many years of programming and development. This may explain why current research has only produced domain-limited prototypes of ITS that concentrate on specific tutoring problems ([Sel88], [Neu89]), but the eventual benefits for language instruction remain obvious.

1.3.3 Microworlds

Intelligent microworld systems implement a conversation with the computer that is limited to a discussion of some constrained "world" or domain. The essential difference between a microworld system and a dialogue program lies in the subject of the simulated conversation: microworlds are restricted to a highly controlled context. This leads to a high degree of predictability with respect to the vocabulary and structures to be encountered, which, in turn, greatly facilitates the task of parsing and understanding the language used by the student. Moreover, the ease of representing knowledge in such a restricted world allows developers to concentrate their efforts on other problems. In language instruction, this means more time to tackle pedagogical considerations such as the correction of ill-formed or semantically incoherent inputs.

Although the techniques employed in microworld design borrow heavily from adventure games, microworld systems go substantially beyond the rather rigid response mechanisms that such games generally use. They are capable of understanding and reacting to student inputs as well as identifying linguistic errors.

Notable examples of experimental microworld implementations are (1) Winograd's SHRDLU [Win84], a system capable of describing and the locations of coloured blocks,[18] and (2) MIT's Project Athena LINGO system ([Mul89] and [Und89]), a prototype in semantically-based CALI, which can interact with students learning German grammar and understand the meaning of the sentences, in addition to flagging grammatical and spelling errors.

[17] Student models are believed to be essential to effective intelligent CAI, but Self [Sel87, p.84] could only identify twenty or so programs in the literature that make use of these models. "It is clear that there is a wide range of potential uses of student models, that, in many cases the computational techniques are not well developed, and especially that the rationale justifying the various uses is often lacking".

[18] The reader will note that SHRDLU is not a language instruction system, but is mentioned here because it is the first, and a very famous, implementation of a microworld system.

1.3.4 Open-ended Text Processing

Open-ended text processing systems simulate the role of a text editor. Focusing on the syntax of the language, such systems perform an analysis of written texts to point out errors, but they make no attempt at teaching the principles they correct. Most open-ended text processors have concentrated on grammar and are often referred to as *intelligent grammar checkers* (IGC). Slightly more general-purpose systems have also been developed to process the input text with respect to style and diction as well as grammar. As these systems are much related to the present work, they are discussed more thoroughly in Chapter 2.

As instructional tools, open-ended text processing systems can handle significant language analysis problems, such as subject/verb agreement, and verb tense formation, but individual systems have demonstrated varying levels of sophistication in terms of the number and type of errors they find, as well as in their ability to find those errors.[19]

Each of the system types described above uses AI techniques in natural language processing and knowledge representation for language instruction. However, the use of such techniques alone does not guarantee that programs will be beneficial for teaching. It is necessary to focus on areas where AI techniques will tackle important aspects of language teaching with greater success than is possible with simple, unintelligent programs, or even through classroom instruction.

1.4 An ITS for Teaching Writing

Among the types of ICALI systems described in section 1.3, the prototype developed in this thesis (STASEL) follows the philosophy of intelligent tutoring systems, and has been designed to fit the architecture of an eventual ITS for teaching writing (Figure 1.2). Based on the ITS model described in section 1.3.2 and on the instructional principles outlined by Neuwirth[20], the proposed architecture encompasses the four components of the model:

1. A natural language processing (NLP) interface that serves as the linguistic component and is composed of a parser, a lexicon and a natural language generator;

2. A pedagogical component that includes the NLP interface and the expert module, as both module contain linguistic information that is relevant to language instruction, operates the explanation planner and determines the tutorial strategy;

3. An expert module that analyzes the grammatical, stylistic, and pragmatic aspects of the input and passes its analysis to the explanation planner;

4. A student model that stores information about the student's performance, and interacts with the pedagogical component, the expert module and the NLP interface.

In the expert module, an intelligent grammar corrector module (IGC) examines the failed parse of ill-formed inputs and corrects the grammatical errors it finds. This ensures that both valid and failed parses are processed by the entire tutoring sequence. Once the grammatical analysis has been performed, the input is analyzed according to principles of style,

[19] See Catt [Cat88] for a detailed assessment of intelligent grammar checkers.

[20] See Section 2.2 for an outline of Neuwirth's instructional principles.

pragmatics and rhetoric[21]. Each sub-module of the Expert Module outputs an analysis that is used by the explanation planner to generate an appropriate and thorough response.

Figure 1.3 highlights the specific contribution of STASEL to the architecture. STASEL, like other works in ITS design, addresses a subset of the knowledge and procedures that constitute intelligent language tutoring. For example, the system cannot plan the content of explanations, nor does it make use of a student model or a natural language generator. The prototype is currently limited to the implementation of the linguistic component (NLP interface) and the stylistic portion of the expert module. As the explanation planner is not yet available, the analysis of the stylistic module is fed to an external *instructional response* component which emulates the generation functions.

[21] Pragmatics considers language in the light of the practical consequences of its use while rhetoric is the study of the techniques which help in making language effective.

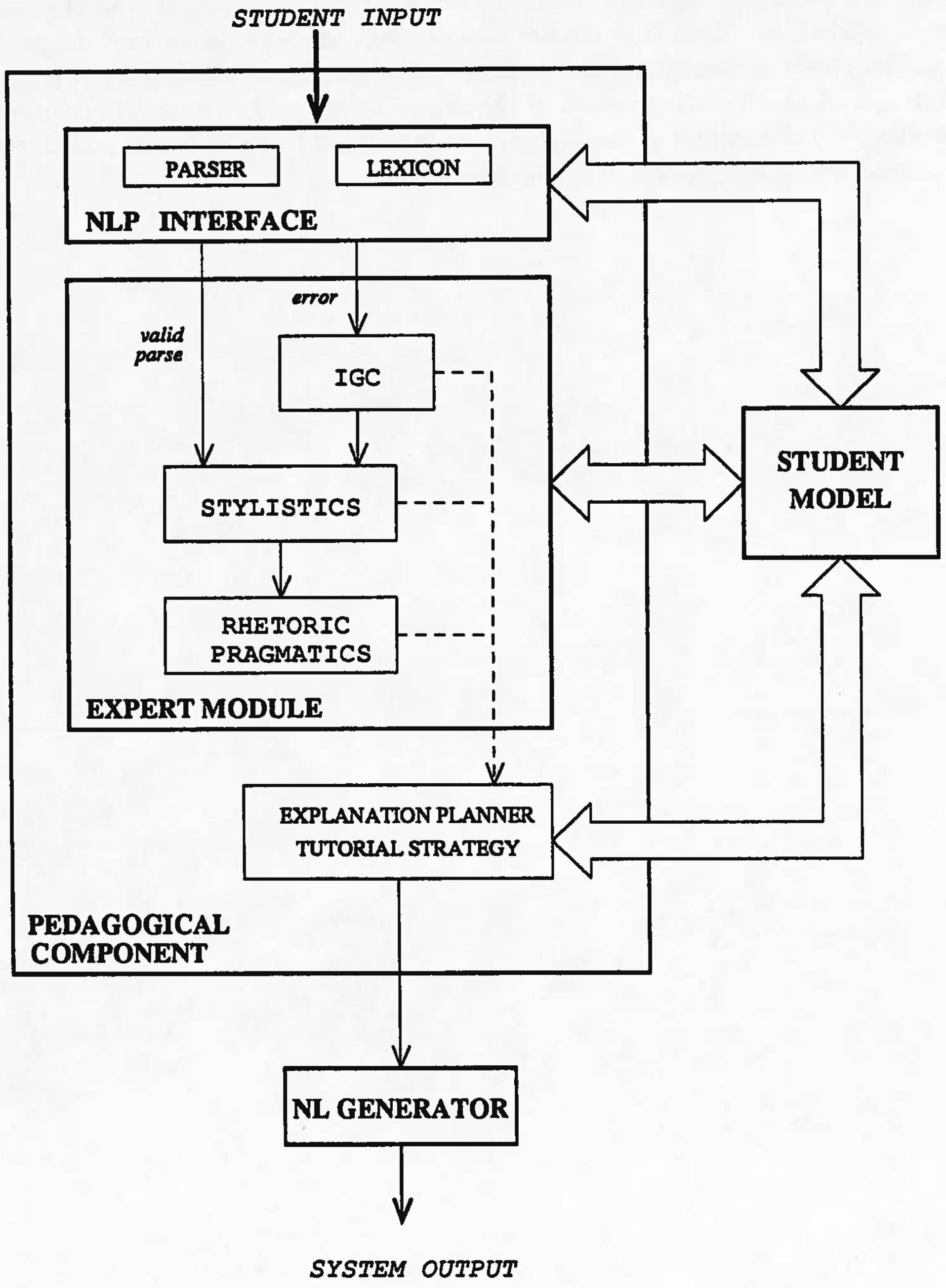


Figure 1.2 An ITS model for teaching writing

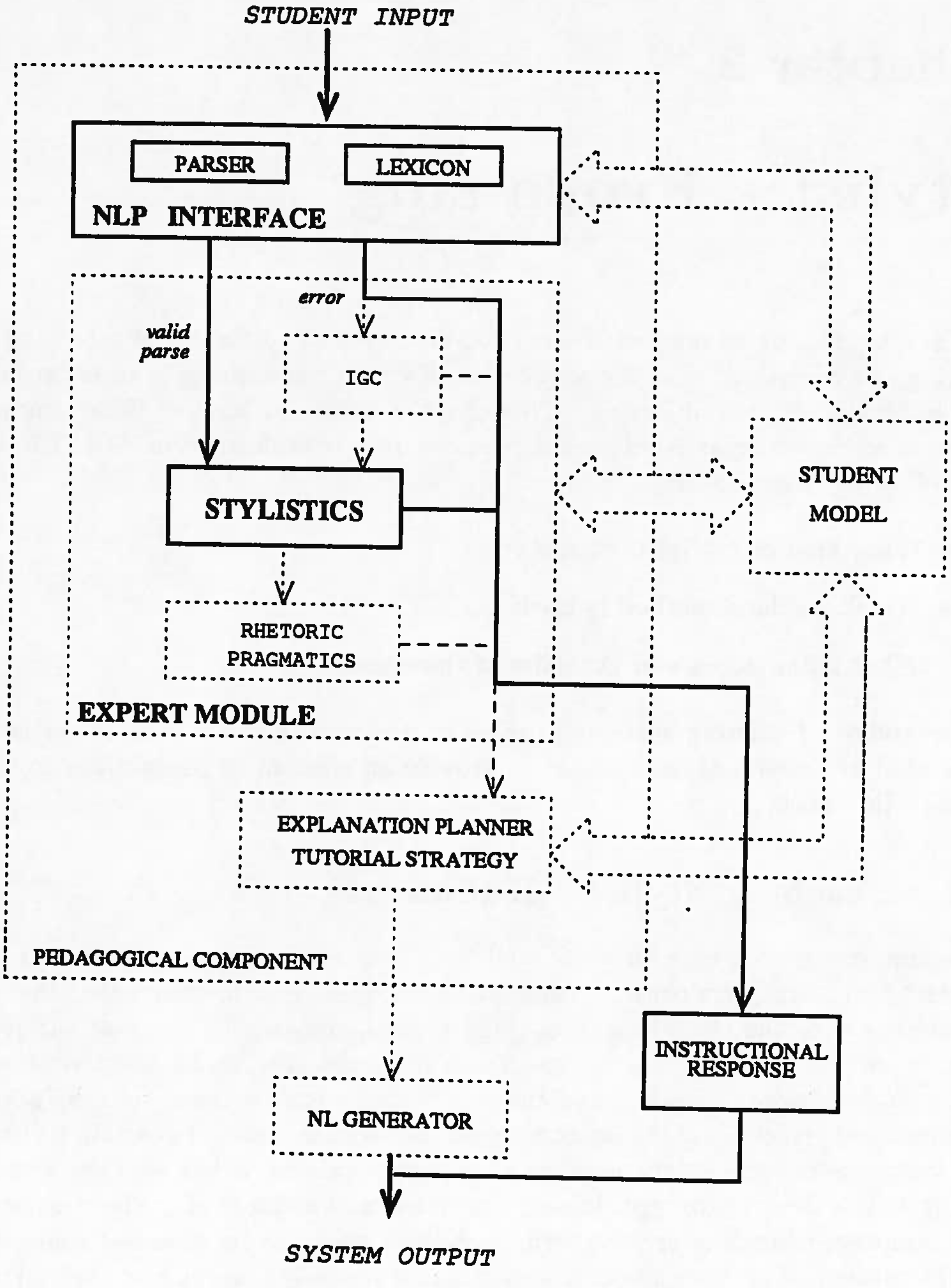


Figure 1.3 The contribution of STASEL to the ITS model

Chapter 2

Stylistic Processing

In the introduction to the first chapter, we discussed the difficulty of achieving good style in writing and contended that the acquisition of stylistic rudiments is an essential step toward the fulfilment of this objective. This chapter describes how stylistic processing for the purpose of computer-assisted language instruction is performed in STASEL, and discusses the following design issues:

- What kind of "style" is taught;
- Which teaching method is used;
- What is the domain of expertise of the system.

An overview of existing systems capable of performing some form of stylistic analysis is presented at the end of the chapter to provide an element of comparison to the techniques used in this work.

2.1 Teaching Stylistic Rudiments

The first step in learning to write well is, of course, a matter of grammar. Generally, a student first learns such considerations as how to spell, how to ensure that the subject agrees in number with the verb, how to conjugate verbs and employ them at various tenses, and how to coordinate and subordinate clauses properly. Yet, to be effective, the components of a sentence (words, phrases, and clauses) must be put together not only according to the grammatical principles of the language used, but also according to certain stylistic principles, for writing is not simply the absence of grammatical errors, but also the presence of *style.*

Style is a flexible concept, loosely restricted and capable of endless variations. Stylists and language educators use the term in various ways. In its broadest sense, "style" refers to the totality of all the choices a writer makes regarding words and their arrangements in order to convey his[1] views. STASEL looks at style as the distinctive, formal, and standard manner of expression characteristic of effective writing. It aims at teaching the norms and conventions (referred to as *stylistic rudiments* from now on) that are accepted, in textbooks on style, as conforming to standards of formal writing. This definition has been extended to

[1] For a thesis so largely based on the conventions of written English, I have chosen to use the conventional masculine pronoun when referring to a generic category of persons.

include the study of the arrangements and the constructions within a sentence that support the realization of specific writing goals.

Two aspects of instruction in stylistic rudiments have been incorporated in this research:

- Teaching *syntactic style* that is, how choices in diction and sentence structure influence the stylistic effect;
- Teaching *goal-directed style* that is, how an intended stylistic goal may be translated into prose.

The first approach considers style in the light of the syntactic and constructions used in the prose. Students learn that shifts in stylistic effect can be created through subtle variations in the fabric of the prose, through careful selection of the words used, and through minor modifications to the syntactic constructions employed. Style is subjective, context-dependent, and therefore difficult to reduce to rules, but stylists and language teachers more or less agree on the principles and standards that make up good syntactic style. One may think of these principles as *rules of style*, even though they do not carry the rigidity of corresponding grammar rules and leave far more room for interpretation. With such rules of style, provided they are discussed in context and never enforced arbitrarily, teachers can effectively communicate stylistic rudiments. In his essay "Teaching Style", Corbett [Cor86] defines stylistic rudiments by separating them into four broad categories called "syntactic variants":

- *Diction*
  The study of the lexical elements of style, or how to use the proper words at the proper places. For example, a slang term may be appropriate in informal communication such as letter writing, but not necessarily in formal writing.

  The cop (*slang*) entered the bank to arrest the thief.
  The policeman entered the bank to arrest the thief.

- *Sentence Construction*
  The study of sentence organization and structure. This category includes the analysis of the location and role of sentence elements, the methods of expanding sentences, the techniques of embedding, the choice of sentence openers, the variety of sentence types, and the study of sentence length and rhythm.
- *Figures of Speech*
  The art of using figurative language to enhance the richness of the prose. It is a deliberate deviation from ordinary patterns and literal meaning.
- *Paragraphing*
  The skill required to assemble a group of sentences all relating to and developing a common idea. The art of good paragraphing is knowing how to organize the sentences, how to link them together, and how to develop the topic of the paragraph coherently.

In *syntactic style* analysis, we refer to *good* or *bad* style: good if the variants chosen by the writer observe the defined conventions of formal writing; bad if they violate them. Teaching the rudiments of syntactic style thus involves flagging errors and providing an explanation as to what constitutes the error, why the error is considered bad style, and how to remedy the situation.

In *goal-directed stylistics*, the second approach to teaching stylistic rudiments, the problem is viewed from the opposite angle: the student is asked to identify first the effect he intends to achieve, and second the structures that will enable him to fulfil his goal. For instance, the goal may be that a student wishes to write clear and concise sentences, as most writing textbooks would advise him. If the student has formal knowledge of the basic structures that induce a sense of clarity to a sentence, the writing process will be simplified. For example, consider the following sentence:

It suddenly began to rain.

If the writer's goal is to *emphasize* the suddenness with which the rain began to fall, a common technique, called "isolation",[2] is to separate the adverb "suddenly" from the core of the sentence, as follows:

Suddenly, it began to rain.

It began to rain suddenly.

The adverb can occur anywhere, but it is most effective if it appears at the beginning or at the end of the sentence, because these positions carry inherent emphasis in themselves. The careful use of the opening and closing positions in a sentence enables the writer to achieve inner emphasis and to communicate nuances in meaning that depend less upon individual words than upon their arrangement.

The use of "isolation" to create emphasis is an example of a goal-directed structural technique, and such techniques can be formally taught as part of a stylistic instruction program. In goal-directed style, the meaning of "style" is more positive and flexible than in syntactic style analysis. One can speak of *inefficiencies* in goal-directed style, but not of *bad* style: "It suddenly began to rain" does not violate any stylistic convention, but is stylistically ineffective if the objective was to emphasize how suddenly the rain began to fall. In such cases, style is understood to include various possibilities of structuring sentences, each appropriate for some purpose, and less so for others. Teaching goal-directed style thus involves recognizing the sentence structure, deciding whether the structure respects the principles associated with the particular writing goal under consideration, and providing a diagnosis that either underlines the appropriateness or pinpoints incongruities in the structure. The objective of the teaching is not to show right answers from wrong ones. Rather, it is to direct student attention to potentially ineffective constructions and offer assistance in understanding and rectifying them.

STASEL both detects errors in *syntactic style* and and recognizes the syntactic constructions that are associated with particular *goal-directed* stylistic features. For demonstration purposes, the scope of the analysis is limited to diction and sentence construction problems, in the first case, and to the goals of clarity[3] and conciseness, in the second.

[2] Repetition, use of position, serial accumulation, predicate adjectives etc. are other common techniques to achieve inner emphasis. [Kan83, pp.295–307]

[3] Clarity is understood to mean clarity *in structure* rather than clarity *in meaning*, as, in the absence of semantic capabilities, STASEL cannot comment on the meaningfulness of a sentence. The system looks solely at the surface structure and bases its analysis on the syntactic elements it recognizes. Consequently, a sentence such as

I swim in the office.

will be judged by STASEL as being stylistically correct and structurally clear as indeed it is, even if it is nonsense.

2.2 Learning by Doing

Researchers in ICALI agree that an ITS should model the role of a teacher as a tutor rather than a lecturer, but they diverge in deciding how one ought to tutor, and which theory of learning and instruction should motivate the design of the pedagogical component of the ITS. This leads to the implementation of various teaching methods in the ITS software that depend on the nature of the subject taught and on the purpose of the instructional program.

For example, some programs, such as Bailin's VERBCON and PARSER [BT88], advocate a traditional teaching method in which the nature and the order of the material to be learned is pre-determined,[4] while others, such as the prototype SWIM developed by Zock et al. [ZLF89], let the student explore the material and realize their errors on their own.[5]

In the design of STASEL, the primary consideration was not to introduce the most current or fashionable learning theory, but to provide a formal computational linguistic basis onto which pedagogical theories could eventually be added. Consequently, the system was built around a flexible pedagogical framework, known as *learning by doing*, which is unrestrictive and open to later revisions by teaching specialists. The concept of "learning by doing" is a theory of instruction which contends that a person can acquire some of the skills necessary to perform a given activity by repeatedly practicing this activity [Neu89, p.49]. However, the theory of "learning by doing" does not prescribe particular teaching guidelines or offer the precise formulation of method required in ITS design. To remedy this lacuna, Neuwirth outlined a series of "learning by doing" principles for instruction, derived from ITS research and readily applicable to computerized models:

- allow free-form inputs or responses;
- provide instruction after each interaction;
- communicate expertise to the student through system responses;
- give immediate feedback on errors;
- use student models to gauge appropriate instruction.

With the exception of the use of a student model, these principles have been incorporated in the design of STASEL. The free-form input requirement is respected in that STASEL accepts a large variety of sentence structures, sufficient to cover the needs of intermediate students, while the lexicon can readily be extended to include an important subset of the dictionary. The system interacts with the student after each input and provides instructional feedback accordingly. This ensures an initial pedagogical support for skill acquisition, which is practical and flexible enough to allow the addition of other theories of instruction.

[4] Such programs are suitable when the subject taught requires a great deal of practice, such as in the acquisition of grammatical skills. They generally follow a set learning sequence that consists in (1) exposing the material, (2) letting the student practice, (3) testing the knowledge gained, and (4) evaluating the performance.

[5] Proponents of learning by exploration (also known as *discovery learning*), as opposed to guided practice, believe that a system built according to unrestrictive principles should not give a student immediate feedback on an error; instead, it should pick and submit salient tasks that will increase the likelihood that the student will notice the error (!).

2.3 Stylistic Treatment at the Sentence Level

> *To understand why anyone writes badly, we have to be able to look at a sentence and understand how it works, how the ideas have been distributed through its different parts, and then decide how to write it better.* [Wil81, p.12]

STASEL performs stylistic analysis at the sentence level. The system accepts complete sentences as input, but does not allow connected text to be entered. As it makes no correlation between successive inputs, STASEL does not have the capability of analyzing the coherency and organization of paragraphs.

Though some would argue that proficiency in the techniques of inner sentence manipulations is not sufficient to ensure that a student write well, sentence level treatment follows from a recognized "divide and conquer" teaching approach: a student should understand and master the basic elements that constitute a written text, notably the sentence, before tackling the more challenging task of putting these elements together. With its explicit and well-defined structure, widely documented in writing textbooks and readily decomposable for computer applications, the sentence offers an ideal medium through which computer-assisted language instruction can be achieved effectively.

Very few CALI programs have so far attempted to tackle the difficult task of integrating stylistic principles in the analysis of sentences let alone attempted to teach these principles for effective writing. The remainder of this chapter reviews some of these programs, outlining the inadequacies and lacunas that prevent their use in a teaching environment and demonstrating where and how STASEL may provide a better alternative.

2.4 Related Research

This section reviews representative research in computational stylistics[6], and evaluates five current approaches with respect to language instruction. Most existing programs concerned with stylistics belong to one of the following categories:

1. *Computer analysis of literary texts*
   Stylistic analysis programs describe and analyze the literary form and content of written texts. These programs focus on quantifiable features, such as the frequency of occurrence of words, vocabulary items, or grammatical forms, and rely on statistical methods to determine the characteristics of an author's work.
   Virtually all stylistic analysis programs use a non-AI computational approach and depend on human interpretation to draw conclusions on the stylistic significance of the data compiled. ([Iri89], [Opp88], [Mil86])

2. *Style-checking facility*
   Style-checking programs serve as a software tool in computer-assisted composition. Available in a variety of forms, from glorified spelling checkers to comprehensive grammatical and stylistic checkers, these programs detect common grammar, diction, and usage errors in texts, and determine quantifiable parameters such as word count and sentence length. Most of these programs are designed for commercial use in

[6] Computational stylistics is the processing of stylistic information using computers.

office environments and have not been widely applied to computer-assisted language instruction.[7]

As the computer analysis of literary texts is more concerned with the statistical evaluation of literary data than with the teaching of style, it is not considered in the present review. Style-checking programs, though not strictly oriented toward stylistic instruction either, demonstrate some instructional capabilities and thus, warrant investigation as potential ICALI applications.

Three style-checking programs are discussed. First, a typical style-checker, the UNIX-based Writer's Workbench [MFK82] is reviewed to give an appreciation of the limitations of conventional, unintelligent style-checkers. Then, two parser-based stylistic processors designed for use in the business community are examined: a commercial program developed by Houghton-Mifflin called CorrecText [Dob90], and the IBM experimental text processing system known as CRITIQUE [RBH88]. Each style-checking system is evaluated according to the following criteria:

- linguistic processing capabilities
- stylistic coverage
- pedagogical content

The last two systems described in this review are special cases. The first system, Neuwirth's PARNASSUS project [Neu89], does not perform any stylistic processing of sentences per se, but it investigates the more general problem of teaching effective writing from the point of view of intelligent tutoring system design, which is a central concern of this work. The second system, DiMarco's STYLISTIQUE [DiM89], implements the foundation of a computational model of goal-directed stylistics and is intended to serve as the basis for a machine translation system capable of preserving style across translation. As we shall see in Chapter 5, DiMarco's model provides the theoretical justification for the design of STASEL's goal-directed stylistic analyzer module.

2.4.1 The Writer's Workbench

The Writer's Workbench [MFK82] is a set of programs designed to aid writers in editing documents. It consists of two major components:

- An automatic proofreading system that searches for spelling and punctuation errors, consecutive occurrences of words, wordy or misused phrases, and split infinitives;
- A stylistic analyzer that describes the writing style of a document, verifies the readability and sentence characteristics, and suggests improvements.

The Writer's Workbench is designed "to help writers produce better written and more readable prose" [CV86]. The developers also claim that the system may be "useful for studying the writing process", though no teaching methodology or pedagogical principles are apparent in the system's design. The stylistic analysis is based on a set of programs

[7] Two of the style-checking programs reviewed in this section, the Writer's Workbench and CRITIQUE, have reportedly been adapted to composition and second-language classes at the university level ([Cat88, p.20], [RBH88, p.196]).

known as the STYLE and DICTION programs: STYLE measures readability[8], sentence and word length, sentence type, and word usage, while DICTION locates common examples of wordy phrasing and bad diction.

As in STASEL, the STYLE and DICTION programs use the term "style" to describe the results of a writer's particular choices among individual words and sentence forms. They limit the stylistic analysis to the sentence level, and assume that sentences are initially well-formed. However, the comparison stops at this point. The Writer's Workbench stylistic portion shows no formal knowledge of the English language, no structural representation of either stylistic or grammar rules, no reference to writer's goals and intentions, and no understanding of the implication of stylistic variations. Its stylistic coverage is limited to some lexical and word usage considerations, and to the flagging of pre-determined wordy phrases.

The most important limitation remains the inability of the system to perform intelligent language processing. To analyze a sentence, STYLE and DICTION use what Dobrin calls a "minimal parsing strategy" [Dob90, p.69] whereby the sentence is broken into its constituents by first referring to a word list that identifies the parts of speech of each word,[9] and second, by using simple pattern-matching to recognize word sequences that correlate with a grammatical error. No attempt is made at parsing the sentence as a whole. The aim is to find syntactic structures that correspond to specified grammatical errors, and parse just enough words to find the necessary correlation.

Such minimal parsing techniques demonstrate insufficient knowledge of the language structures and thus fail to produce a valid analysis whenever sentences become complex or syntactically ambiguous. For instance, STYLE and DICTION have trouble

- distinguishing between an introductory participial phrase, a main clause, and a gerundival subject;
- differentiating between simple, double, and triple noun modification;
- recognizing participles used as adjectives;
- analyzing any complex sentence,

all of which are successfully and correctly understood by STASEL's parser. In addition to the other inadequacies mentioned above, the considerable limitation in linguistic processing capability strongly diminishes the initial appeal of the Writer's Workbench, and disqualifies it as a competent stylistic tutor.

2.4.2 CorrecText

CorrecText is a commercial style-checker produced by Houghton-Mifflin's Business Software division and marketed for the PC under the name CorrectGrammar. The program consists of a parsing component, which decomposes the sentences, and an error correction facility,

[8] Readability is measured by calculating statistical indices based on sentence and word lengths. "Although the indices may not measure whether the document is coherent and well organized, experience has shown that high indices seem to be indicators of stylistic difficulty" [CV86, p.32–3].

[9] STYLE uses a small dictionary of 350 words to assign word classes to part of the text and then uses "experimentally derived rules of word order" to classify the rest of the text "with an accuracy of about 95%" [CV86, p.32–2].

which detects grammar, style, and usage errors. The program also suggests alternatives, and permits the user to make corrections.

Primarily motivated by business and office requirements, the design of CorrecText resembles more that of a grammatical checker than that of a stylistic processor. The system detects mostly non-stylistic problems, none of which were considered a priority in the design of STASEL:

- spelling errors and typos;
- grammatical problems including verb form errors, subject/verb and number disagreement, and sentence fragments;
- punctuation mistakes;
- errors in typographic conventions such as capitalization and the use of numbers.

Its stylistic component covers word usage standards and context-free writing practices of the kind found in the appendix of most writing textbooks: split infinitives, double negatives, colloquialisms, informal habits, wordy phrases and redundancies. In this sense, CorrecText functions much like STASEL, but it bases its analysis on an unsystematic and informal treatment of style. Moreover, it has no sense of goal-directed stylistic analysis and shows no teaching disposition. It is solely aimed at relieving business writers from the task of refining documents by getting rid of spelling errors, common grammatical mistakes, and prescriptive stylistic misusages.

The interest of the program lies in its use of a parsing component that yields fewer incorrect parses and makes more complex sentence analyses than pattern-matching approaches of the Writer's Workbench sort. Since the techniques used in the parser's design are proprietary, only the account provided by Dobrin [Dob90] can shed light on the system's operation and performance. An inspection of the examples that Dobrin fed to the system reveals important flaws in the parsing mechanism, which ultimately make it unsuitable for language instruction purposes. CorrecText uses a probabilistic parser (referred to as a *full* parser by Dobrin) that does minimal semantic analysis and has no backtracking capabilities.[10] It seemingly resolves syntactic ambiguities during the parsing process by applying stochastic clues derived from "an analysis of how errors actually occur". In dictating decisions to the parser, these clues are meant to increase the probability of catching grammatical errors, but, at the same time, this use seriously threatens the correctness of the resulting parse. In fact, Dobrin reports several instances of incorrect syntactic analyses. For instance, the following example sentence (p.69, example 2)

> The Greek Islands form a barrier to progress through the Aegean sea.

is wrongly diagnosed by CorrecText as being an incomplete sentence (fragment).

Frequent incorrect parses disqualify CorrecText from any potential language instruction application, as such errors cannot be tolerated in a teaching environment. The use of a non-stochastic parser, such as the one implemented in STASEL, is essential to ensure a high degree of precision in the syntactic analyses, which, in turn, will contribute to a more effective and adequate instruction. Dobrin acknowledges this limitation when he comments on the usefulness of CorrecText's probabilistic parser:

[10] A distinction is drawn between a *probabilistic* (or full) parser and a *rule-based* parser of the kind implemented by STASEL. In the former, the use of probabilistic clues for sentence disambiguation may lead to incorrect parsing, whereas the latter aims at producing a unique correct parse, or no parse at all.

> A full parser cannot be made into a [non-stochastic] parser. The methods that allow a program to uncover accurately the maximum number of grammatical errors are not necessarily those that are useful in second-language training. (p.71)

2.4.3 CRITIQUE

The only other style-checking program that analyzes sentences using a parser is an experimental program called CRITIQUE. Born as an extension of the EPISTLE project [HJM$^+$82] begun at IBM in 1980, CRITIQUE is a prototype system that identifies grammatical and stylistic errors in English texts. It was developed for use in three major application areas: office environments, publication organizations, and educational institutions.

Operating in conjunction with a word processor, CRITIQUE parses the text entered and presents the user with a "critique" that points out the types and sources of the errors encountered. The processing of the text is performed in six steps:

- **Step 1** determines the sentence and paragraph boundaries of the text;
- **Step 2** uses a 100,000-entry dictionary and hundreds of syntactic rules to identify spelling errors, non-existent words, and awkward phrases;
- **Step 3** performs the syntactic parsing, detects grammatical errors, and produces a parse tree representation of the text;
- **Step 4** diagnoses potential stylistic problems such as incorrect word usage, split infinitives, redundancies, over-qualification of nouns, and ineffective phrase constructs;
- **Step 5** generates statistical information about the document on the basis of the lexical and syntactic analyses;
- **Step 6** organizes the output format, summarizing the errors found and tabulating the statistical information.

CRITIQUE also provides three levels of on-line help: the first level simply identifies the error, the second provides a brief explanation, and the third offers a more detailed diagnosis. In tests, the system has shown to be capable of handling a variety of different types of sentences and of recognizing a wide range of grammatical errors [Bai88].

In many respects, CRITIQUE may appear to closely resemble STASEL.[11] Both use a backtracking parser, treat input in a modular sequence, can recognize complex syntactic forms and various sentence types, can flag typical errors in syntactic style, and give a corrective explanation after each error. The systems bear little resemblance, however, when it comes to the specifics of their respective stylistic processing approach.

In CRITIQUE, the development of stylistic critiques is largely normative ("prefer the active voice, use strong verbs, cut down on intensifiers") and tailored to a business environment rather than a classroom. The stylistic analysis is performed with an unstructured mass of rules that mostly serve either to compute quantifiable parameters, such as sentence length or the number of passive verbs, or define readability thresholds that indicate whether

[11] The comparison is, of course, relative to functionality rather than overall performance. Lacking the several years of development put into CRITIQUE, STASEL cannot reasonably be compared with some of the more laudable non-stylistic elements of the IBM system, notably its powerful grammatical correction component and its sophisticated user interface.

a normative principle has been violated. Threshold violation is determined in what appears to be a purely mathematical, non-linguistic, fashion: "Agreed-upon principles of good business style" [HJM+82, p.232] are broken down into classes of violation (e.g. too many words in a sentence, too many dependent clauses, too great a distance between the subject and the verb). Each class of violation is expressed as a *critique function*, a mathematical function of specially identified grammatical "cues", each carrying a weighted value.[12] "A primary criterion for selecting the function is that the value of the function should increase more or less linearly with perceived badness of style" [HJM+82, p.233]. Each critique function is evaluated to a single number that is compared to pre-determined system thresholds that can be fine tuned by the user. If the value of the function exceeds the threshold of a particular stylistic principle, a violation is reported and a critique generated.

The use of thresholding for stylistic processing may yield a readability interpretation that is satisfactory in a business environment, but it is unsuited to stylistic instruction. The technique is too coarse-grained and shows too little consideration of the underlying syntactic structure to be of any use in teaching subtleties of language. Even though Richardson and Braden-Harder [RBH88] have suggested that CRITIQUE could be used as a CALI tool in language classes, the system's stylistic treatment lacks the necessary formal understanding of how syntactic structures affect the stylistic shape of a sentence to perform adequate goal-directed stylistic instruction.[13]

2.4.4 PARNASSUS

Even though PARNASSUS concentrates on language processing at the paragraph level rather than specifically targeting sentence-level stylistics, the system shares a design objective of STASEL in focusing on the problem of teaching students how to write effective sentences, appropriate to the context in which they are written. Unlike STASEL, however, which concentrates on the implementation of a thorough linguistic component that is used to feed the stylistic portion of the expert module of the ITS, the design of PARNASSUS emphasizes the pedagogical considerations as well as the rhetorical portion of the expert module. In Section 2.2, we used the pedagogical principles that were outlined for PARNASSUS to give STASEL a basic tutorial strategy, but the design focus chosen by Neuwirth, which addresses rhetorical problems without the support of a strong linguistic component, severely restricts the linguistic processing capabilities of the prototype and may eventually force Neuwirth, in her own opinion, to rethink her original strategy.

PARNASSUS is designed to provide students with some understanding of the concepts of paragraph coherency and emphasis.[14] The student is asked to read a paragraph that lacks coherence and emphasis, and to produce a revision of it in which the sentences have been made more effective by choosing alternative syntactic patterns. (The exercises are designed so that syntactic strategies suffice to fix the paragraph problem.) PARNASSUS compares the student's revision to a set of possible revisions previously "rated" according to a model of

[12] For example, the violation class *too great a distance between the subject and the verb* has two grammatical cues: the location of the subject and of the verb, identified as *Loc(Subj)* and *Loc(Verb)*.

[13] The developers have recognized the difficulty of porting CRITIQUE to an educational context, acknowledging that the "use by educational institutions has proven to be the most challenging of the three [application] areas" [RBH88, p.196].

[14] In constructing a paragraph, a writer needs to emphasize the main point of the paragraph and ensure that the supporting sentences of the paragraph all fit together in a coherent relationship. See Hefferman [HL82, p.108–111] for more detail on paragraph emphasis and coherency.

the rhetorical goals characteristic of discourse strategy. The comparison is performed by simple pattern-matching.

Incapable of "understanding" the student revision let alone generating a formal representation of its content, PARNASSUS has to simplify the analysis by constraining the student's response and limiting its coverage to a small set of possible sentence structures. Neuwirth acknowledges that parsing techniques, rather than pattern-matching heuristics, may be necessary to enhance the scope of the program [Neu89, pp.55–56].

2.4.5 STYLISTIQUE

STYLISTIQUE is a stylistic parser, written in PROLOG, that interprets the English and French syntactic stylistic grammars developed by DiMarco in her PhD thesis [DiM89]. These grammars provide a theoretical foundation for a computational model of stylistics intended for machine translation and are implemented in separate stylistic parsers in the system. The parsers' rules are used to perform a syntactic analysis of an input sentence and to build up a parse tree representation of its syntactic and stylistic structure. Once the sentence has been completely parsed, another set of rules is applied to the resulting stylistic structure to produce a goal-directed analysis of the sentence. STYLISTIQUE recognizes three different classes of stylistic goals:

- clarity and obscurity
- abstraction and concreteness
- staticness and dynamism

Although STYLISTIQUE's parsers can usually run unaided, some pre-processing of the input sentence, in which explicit punctuation markers (called *pseudo-cuts*) are inserted to exert control on the parsing process, has shown to be necessary to avoid the generation of incorrect parses in the case of complex or ambiguous inputs. The use of pseudo-cuts may be warranted for a machine translation application, but cannot be tolerated in a language instruction environment, as students cannot be expected to know where and how to use them correctly. This limitation prevents Stylistique's parsers from being directly applicable to language instruction purposes, but the theory and principles involved in DiMarco's design remain of a great practical interest.

It is the capability of processing inputs according to goal-directed principles of stylistics that makes STYLISTIQUE pertinent to the present research. After developing a vocabulary of style, DiMarco defined an explicit set of rules that correlate the syntactic patterns of a sentence with specific stylistic goals. These rules represent a significant advance in the stylistic interpretation of syntactic structures, which was previously carried out only by means of examples or simple syntactic rules. Yet more importantly, DiMarco's formal definition of "goal-directedness" can readily be decomposed into computer heuristics and thus applied to the teaching of goal-directed principles in stylistic instruction. This potential application, now a reality in STASEL, was foreseen by DiMarco when she wrote that

> what has been learned from developing STYLISTIQUE could also be applied to machine-aided language instruction. Existing language-teaching systems focus almost exclusively on the basics of composition. An instructional version of STYLISTIQUE could systematically develop a student's understanding of the more advanced aspects of language composition. [DiM89, p.177]

Chapter 3

STASEL: An Intelligent Stylistic Processor

This chapter describes the design and implementation of STASEL, a prototype ICALI system for stylistic instruction. The following sections illustrate the operation of the system and provide a detailed description of its output capability and performance. The features that contribute to STASEL's instructional relevance are emphasized throughout the discussion. While this chapter outlines the computational characteristics of the system, the details of its stylistic processing capabilities are the subject of a separate discussion presented in Chapters 4 and 5. The examples referenced in the next sections, if not written out in the text, can be found in Appendix B.

3.1 System Design

STASEL is a PROLOG-based interactive system capable of analyzing the syntactic and stylistic features of sentences for the purpose of language instruction. It consists of 4,400 lines of code, more than 500 grammatical and stylistic rules, 100 definitions and descriptions, and a 800-word lexicon. The design structure of the system consists of six functional modules (see Figure 3.1):

1. syntactic parser
2. lexicon
3. utilities
4. syntactic style analyzer (SSA)
5. goal-directed style analyzer (GSA)
6. information module containing definitions and explanations

PROLOG was chosen over other programming languages because of its strong affinity with natural language processing. Its *descriptive*[1] nature favours the implementation of a knowledge base in which objects (words) and the relationship between objects (grammar or style

[1] The *descriptive* nature of PROLOG contrasts with the *prescriptive* nature of conventional programming languages. The PROLOG approach describes known facts and relationships about a problem rather than prescribe the sequence of steps taken by a computer to solve a problem.

rules) can be formally defined. More practically, PROLOG has built-in pattern matching routines that are particularly useful when searching for strings of words within a sentence, and a powerful list processing capability that allows a sentence to be stored as an ordered list of items (the words) that is easily manipulated and referenced. Moreover, the coding of the grammar rules needed to parse a sentence is greatly simplified by using a higher-level syntax known as *definite clause grammar* notation [PS87]. This notation is provided by the C-PROLOG compiler and especially developed to help in the design of parsers. PROLOG also displays intrinsic modularity, which is suitable not only for building prototype systems quickly, but also for adding successive stages to an existing program.

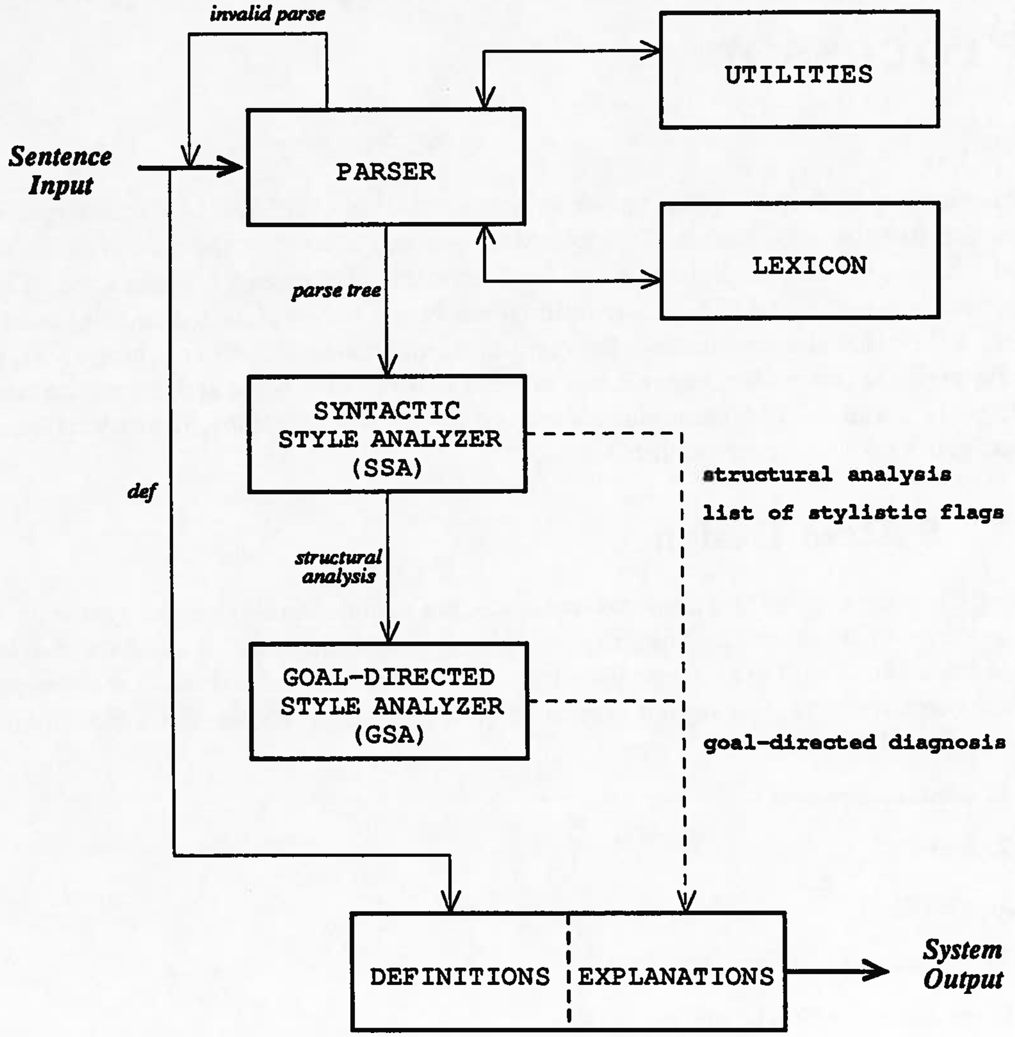


Figure 3.1 STASEL's design structure

3.1.1 System Operation

STASEL processes input in two modes: a definition mode and a sentence analysis mode. The student indicates to the system that a definition is requested by typing the key word **def** before the definition title. Definition requests are not parsed. They are directly passed to the information module where the definition is pulled out from the data base and printed on the screen.

In sentence analysis mode, each input sentence is read and parsed, but only those free of grammatical and spelling errors are allowed to proceed to the stylistic analyzer modules. Hence, the parser, lexicon, and utility modules serve as a front end for the subsequent stylistic analysis. Using the utility module, the parser reads in the input sentence and stores the words in a list structure that is examined from left to right, with frequent backtracking and lexicon lookups, until a parse is produced, or until all the structural combinations known to the parser have been attempted. If the parser fails to find a valid analysis of the sentence, STASEL stops processing and prompts the student to enter a new input. Otherwise the parser produces a structured representation of the sentence, called a *parse tree*, that serves as a basis for the stylistic analyses to follow.

Two independent modules analyze the sentence stylistically. They are the syntactic style analyzer (SSA) and the goal-directed style analyzer (GSA). The syntactic style analyzer module decomposes the parse tree into a higher-level structural representation (called *structural analysis* in Figure 3.1) that is suitable for display to the student, and looks for the presence and, in some cases, the absence of stylistic problems. The result of the SSA stylistic analysis is summarized in a list of stylistic flags that is processed by the information module and printed with the structural analysis as part of the system output. The goal-directed stylistic analyzer uses the structural analysis generated by the SSA module to judge the clarity of the input sentence. Its diagnosis is also included in the final output message.

The information module contains the definitions and explanations that are displayed to the student in the output message. This module effectively "translates" the output of the two stylistic modules into an adequate instructional response. All of the pedagogical feedback available in STASEL is contained in the information module. This modularity allows for easy modification of the message's content and provides a convenient means by which to tailor the instruction to a particular teaching method.

3.1.2 Parsing in STASEL

The parser is a central component of the design of STASEL. It deserves consideration not only because it allows the student to interact with the system using free-form input, an important requirement in a language teaching environment, but also because it is tailored for stylistic processing. The objective of the parser is to break the sentence into its lexical and syntactic constituents according to the rules of the English language. STASEL's parser has several computational characteristics that enhance its operation and provide a comprehensive basis on which to build the stylistic component:

1. it analyzes an important range of English sentence constructions, an essential feature if any form of stylistic analysis is to be worthwhile;

2. it decomposes the sentence into its grammatical and lexical components using integrated disambiguation heuristics to determine in the case of ambiguous constructions which reading is most plausible;

3. its parse tree representation contains information on the stylistic impact of certain words (drawn from the lexicon) and on the stylistic relevance of certain constructions (recognized during the parsing process). This information is made available to the stylistic modules so that detailed feedback on stylistic errors can be produced.

Parsing, in STASEL, involves three on-going processes:

- A *lexical process* that associates each lexical item with its function within the sentence, deciding which word is a noun, an adjective, a verb, etc.
- A *syntactic process* that groups a set of words according to its grammatical contribution to the sentence. This involves identifying each word group as a syntactic structure (noun phrase, relative clause) and each syntactic structure as a functional entity (subject, complement, modifier, etc.).
- A *stylistic process* that associates stylistic features to certain elements found during the lexical and syntactic processes. These features are used to identify the special characteristics of certain words (usage, vagueness) and certain phrases (inverted structure, passive construction, split infinitives) that are relevant to the stylistic analysis to follow.

These combined parser processes produce two independent parse trees: a *syntactic* parse tree that holds a detailed description of the lexical and structural features of the sentence, and a *stylistic* parse tree that provides a more concise account of the structural elements, but gives a precise outline of the stylistic features found during parsing. Only the stylistic parse tree is passed to the following stages of the system.

The Lexical Process

A lexical process is initiated every time the parser tries to recognize a lexical item (word) by looking up the English vocabulary stored in the lexicon. Each lexicon entry associates a word item with its syntactic and stylistic attributes. For example, the lexicon entry for the noun *cop* consists of the singular and plural form of the word, and of a list of stylistic attributes that, in this case, contains only one item that tells the system that "cop" is a slang term.

```
noun(cop,cops,[slang=cop]).
```

Lexical entries are as elaborated as required by the function of the item they describe. For verbs, lexicon entries contain four different attributes: a list[2] of possible complements when the verb is used in the active voice, a list of possible complements for the passive voice, a description of the various forms of the verb, and a list of stylistic attributes.

```
verb([[obj],[that],[none]],
     [[pp(to)],[that],[none]],
     know,knows,knew,known,knowing,
     [attitude=know]).
```

[2] Lists in PROLOG are represented by square brackets holding items separated by commas. [a,b,c] is a list containing three elements a, b, and c. An empty list is simply [].

The Syntactic Process

The syntactic process uses feature-augmented phrase structure rules to reproduce the grammar of English and parse a sentence. A phrase structure rule is a simple statement that indicates how words can be grouped together to form a grammatical entity. For instance, a *noun phrase*[3] in English may consist, among other possibilities, of a proper name, a pronoun, or a noun preceded by a determiner. The phrase structure rules for such a noun phrase would then be expressed as follows:

```
np → proper.
np → pronoun.
np → det,noun.
```

The most important entity in STASEL is the sentence. Each sentence consists of sub-levels of grammatical entities, organized in a hierarchical fashion starting from the lexical items to the complete sentence. This organization can be described in a tree representation of the sentence's components. Consider the sentence *John bought a book* (Figure 3.2). This sentence consists of two grammatical entities: a noun phrase (the subject) and a verb phrase (the predicate). These entities are broken up into their constituents[4] until the words are recognized, as shown in the tree representation. The phrase structure rules and lexicon entries necessary to parse the sentence, and stripped of attributes for simplicity, are the following:

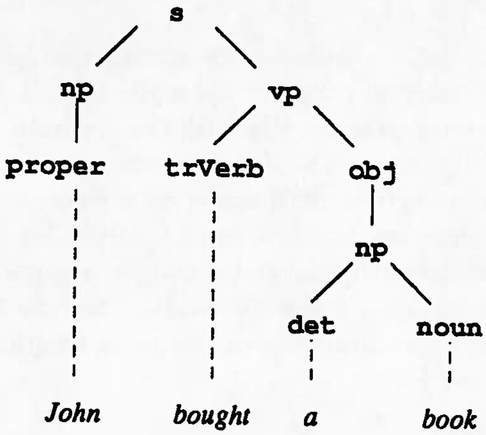


Figure 3.2 Tree representation of *John bought a book*

[3] Definitions of frequently used grammatical terms such as *noun phrase* are available in the glossary of appendix A.

[4] In the phrase structure rules example of *John bought a book*, s stands for sentence, np for noun phrase, vp for verb phrase, trVerb for transitive verb (a verb that takes a direct object as complement) and obj for direct object.

Parser rules	Lexicon entries
s → np,vp.	proper → [john].
np → proper.	det → [a].
np → det,noun.	noun → [book].
vp → trVerb,obj.	trVerb → [bought].
obj → np.	

In STASEL, each phrase structure rule is feature-augmented, that is, accompanied by attributes, such as number, person, case, voice, etc., that further characterize the rule. These features are used during the parsing process to enforce the grammatical requirements of the language. The analysis resulting from the syntactic parsing process is summarized in a structured representation of the sentence, the syntactic parse tree, built during parsing with the help of the utility module. This tree only has an internal relevance to the system and was used in development to test the operation of the parser and to ensure that the structural analysis was correct.

The Stylistic Process

The only function of the stylistic process is to construct a *stylistic* representation of the input sentence for use in a subsequent stage by the stylistic modules. This representation, called a stylistic parse tree, is built concurrently with the syntactic parse tree and thus follows the same parsing sequence. The structure of the stylistic tree, however, is very different from the corresponding syntactic structure. It is based on a *frame statement* notation derived from a semantic structure representation developed by Hirst for his Absity semantic interpreter [Hir87]. Each grammatical entity defined during the syntactic process is associated with a frame statement headed by a frame determiner and containing two elements: a frame name and a list of attributes consisting of slot pairs or other frames[5]. A frame statement is written in the following form:

```
Fdet(Fname,AttList)
```

where `Fdet` is the frame determiner, `Fname` is the frame name and `AttList` is the list of attributes. For example, the frame statement for the noun phrase *the cop* is

```
style(np,[slang=cop])
```

There are two possible types of frame determiners:

`struct` if the grammatical entity is a sentence;
`style` for all other entities that are not sentences.

Frame names identify a grammatical entity and cover the entire range of structures recognized by STASEL's parser. Examples of possible frame names are:

- "struct" frames:
  - `simple` for simple sentences;
  - `compound` for compound sentences;

[5] The terminology used to describe the stylistic parse tree is also borrowed from Hirst's Absity representation [Hir87].

  - inter for interrupted sentences;

- "style" frames:
  - np for noun phrases;
  - active for active verb phrases;
  - desc for adjectival phrases;
  - complex for relative and embedded clauses;
  - gerund for gerundival clauses;

The list of attributes describes the stylistic features of the grammatical entity to which it is associated. If no relevant attribute is detected, the list remains empty. An attribute can either be

1. a frame, as it is the case in the attribute list of a simple sentence frame that holds a noun phrase and a verb phrase frames;
2. a slot pair that either identifies the stylistic feature of a particular lexical item (as in slang=cop) or describes any other pertinent stylistic characteristics found during the parsing process, such as the presence of complements or modifiers in the predicate, the detection of split infinitives, and the number of adverbs.

The grammatical entities that make up a sentence are thus expressed as "style" frames and are combined into a "struct" frame to represent the complete sentence. A "struct" frame can be embedded in another "struct" frame, as it is the case in complex and compound sentences. The resulting frame structure representation becomes the stylistic parse tree of the sentence. The stylistic parse tree of *John bought a book* illustrates the relationship between the various elements of the frame structure and the sentence's components:

```
struct(simple, [
                style(proper,[]),
                style(active,[
                              comp=style(np,[])
                              adv=0])])
```

John, the subject, is represented by a "style" frame named proper and has no attributes, while *bought the book*, the predicate, is represented by another "style" frame named active and lists two attributes: a complement (noun phrase frame) and no adverbs. The entire sentence is contained in a "struct" frame named simple that has the subject and predicate "style" frames as attributes.

The advantages of using frame statements to represent the stylistic structure of the input sentence are numerous. Frames provide a readily accessible structure that not only preserves the hierarchical organization of the sentence but presents the information concisely. Thus, only the information that is relevant for stylistic processing is included in the frame representation. The frame statement notation has greatly simplified the development of the SSA module and reduced the number and size of the rules needed to evaluate the stylistic parse tree of the sentence.

3.2 Output Capability

STASEL is designed to process two kinds of input: a sentence to be analyzed or a request for definition help. To request information about definitions, the user types the key word def followed by the definition title.

```
--> def passives

DEFINITION OF: passives

Style can be made more direct, vigorous, and concise if the writer
avoids unnecessary passive verbs. There are occasions when passives
are legitimate, but if used in excess, they slow down the rhythm of
the sentence. Passives are useful to:
   * help keep the focus of the sentence on someone or something
     important that is acted upon
   * describe an action when the writer does not know or does not
     care to say who or what performs it
   * put the subject of the action at the end of the clause
     where the writer can easily attach a long modifier
```

If the user types def help, the system lists all available definition titles.

```
--> def help

definitions available for:
-------------------------
   abstraction          anticipatory construction
   archaic              centroschematic
   cliche               compound structure
   colloquial           concordant
   deadwood             discordant
   interruption         jargon
   modifying element    monoschematic
   noun phrase (np)     parallel construction
   passives             prepositional phrase (pp)
   pompous              positive interruption
   resolution           slang
   split infinitives    taboo
   vacuous word         vague word
   verb phrase          weak word
```

Otherwise, any other input will be considered as a sentence to be analyzed. Once the sentence has been typed, the parser takes over and attempts to recognize the structure of the input. Its first task is to check for spelling errors and unrecognized words. If the sentence contains an error[6], the system stops processing the sentence, prints the item causing trouble, and requests a new input:

[6] As no spelling-checker has yet been implemented, STASEL does not distinguish between words that are incorrectly spelled or that do not exist, and words that have a proper English meaning, but are not included in the lexicon. Any unrecognized string is perceived as an error that stops any further processing.

```
........................................................
--> The copmuter buffer is empty

  >>>>> The word ..copmuter.. is not in the lexicon

........................................................
-->
```

Once all words have been checked, STASEL's parser analyzes the sentence according to the grammar rules of the parser. If a grammatical error is found, the system suspends all processing and informs the user of the presence of a grammatical mistake. As no grammatical corrector module exists in the present implementation, no attempt is made at defining the source of the error, nor at correcting the problem. However, STASEL is quite rigorous in its grammatical error detection, disallowing sentence fragments, subject/verb disagreement, prohibited verb transitivity, and, as shown below, incorrect pronoun case choice[7]:

```
........................................................
--> She came with John and I

  >>>>> No grammatical parse for this sentence

........................................................
-->
```

The parser gives control to the stylistic processing part of the system only if a valid parse tree has been produced. Obviously, stylistic errors and problems are allowed. After the analyses of the SSA and GSA modules are completed, STASEL outputs a four-segment message stating the result of its findings. The first sentence of this thesis was submitted to the system to illustrate this output capability (Example 3.1). STASEL judged it to be a little more wordy than necessary, but structurally clear. The four segments are clearly shown in the output message: The first segment simply restates the object of the analysis. The second segment gives the result of the syntactic style analysis performed by the SSA module. Every time a stylistic principle is violated, the SSA records a flag and adds it to a list. The information module evaluates the list in a subsequent stage, converting each flag to an instructional reply and printing each reply with a heading that indicates the type of violation encountered. In Example 3.1, two flags, both headed by `[W]`, were printed, suggesting a wordiness problem. The meaning of different stylistic flags and headings is discussed in detail in Section 4.

The third segment displays the result of the goal-directed stylistic analysis. STASEL judges the structural clarity of a sentence according to precise goal-directed stylistic features (see Chapter 5). If the sentence does not respect these clarity features, the system produces a "sentence is unclear" message and states the reason for this diagnosis. Otherwise, the system informs the user that the sentence is structurally clear and prints the name of the stylistic feature of clarity that was recognized during the analysis. The last segment displays the detailed structural analysis generated by the SSA module and used in the GSA module to produce a goal-directed analysis.

[7] Being the nominative form of the first person singular pronoun, *I* can only be used in the subject position. When the pronoun refers to the object of the action, the accusative form *me* must be used.

```
-----------------------------------------------------------------

ANALYSIS OF:  It is a widely accepted view that learning to write
              a language well is a long and arduous process.

-----------------------------------------------------------------

 [W] anticipatory construct introduced by "it"

 [W] restrict the use of vacuous words such as "view"

-----------------------------------------------------------------

 <=> SENTENCE IS CLEAR: monoschematic
-----------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: simple noun phrase (np)
        comp: simple noun phrase (np)
        comp: -- relative clause --
                  clause -->
                      subj: gerundival with infinitive clause
                      comp: simple noun phrase (np)

-----------------------------------------------------------------
```

Example 3.1

STASEL can also display, upon request, the intermediate parse tree representations generated by the parser. The syntactic and stylistic parse trees are not included in the output message to the student, but are made available to the user as they may be of use when learning the particularities of English structure. A single-letter input request may be typed after a given sentence analysis to obtain the desired intermediate tree representation:

p — to print the syntactic parse tree (Figure 3.3)[8];

s — to print the stylistic parse tree (Figure 3.4);

Examples of STASEL's Parsing Ability

One of the strengths of STASEL is its ability to parse an important range of sentence structures. The following represents only a subset of the possible variations allowed by the system:

- sentences with gerundival subject and noun modification (Examples 5.21, B.8);
- sentences with two complements (Examples 4.3, 5.3, B.11);

[8] Figure 3.3 has been edited to highlight the location of the words within the parse tree.

```
s(
  subj(
      np(
          head(pron(it)),
          num(singular),
          pers(third))),
  mainVerb(
      vp(
          verb(be,present),
          form(finite),
          num(singular),
          pers(third),
          (obj(
              np(
                  det(article(a)),
                  adjG(adverb(widely),participle(accepted)),
                  head(noun(view),singular)),
                  relSent(
                      relpron(that),
                      s(
                        subj(
                            vp(
                                verb(learn),
                                form(presP),
                                num(singular),
                                pers(third),
                                obj(
                                    infinitive_np(
                                        prep(to),
                                        verb(write,infinitive),
                                        form(nonfinite),
                                        num(singular),
                                        pers(third),
                                        obj(
                                            np(
                                                det(article(a)),
                                                head(noun(language),singular)))),
                                        vpMods(adverb(well)),
                                        voice(active)))))),
                                voice(active))),
                        mainVerb(
                            vp(
                                verb(be,present),
                                form(finite),
                                num(singular),
                                pers(third),
                                obj(
                                    np(
                                        det(article(a)),
                                        adjCL(
                                            adj(long),
                                            conj(and),
                                            adj(arduous)),
                                        head(noun(process),singular)))),
                                voice(active))),
                        type(simple))))),
        voice(active))),
 type(simple))
```

Figure 3.3 Syntactic parse tree representation of Example 3.1

```
SUBJECT:
  style(pron,[att,ant=it])

VERB PHRASE:
  style(be,[
        ant=be
        obj=style(np,[
                 adj=1
                 adv=1
                 filler=view])

        obj=
          ................................
          SUBJECT:
            style(gerund,[
                     comp=style(inf,[
                               comp=style(np,[
                                        adv=1
                                        mod=adverb])])])

          VERB PHRASE:
            style(be,[
                  ant=be
                  obj=style(np,[
                           adj=2
                           size=long
                           subst=process])])
          ................................])
```

Figure 3.4 Stylistic parse tree representation of Example 3.1

- sentences with several levels of nested prepositional phrases (Examples 4.6, 5.9);
- passive sentences or passive constructions (Examples 4.5, B.4, B.6);
- sentences with parallel three-way noun conjunction in the complement position (Example B.7);
- complex sentences with
  - several levels of embedded clauses (Examples 5.6, B.10, B.17)
  - subordinate clauses (Examples 5.8, B.19);
  - infinitive initial modifying elements (Examples 4.5, 5.13)
  - adjectival final modifying elements (Example 5.10)
- compound sentences (Examples 5.18, B.1);
- sentences with two coordinated verb phrases (Examples 5.19, B.10);
- sentences interrupted by
  - appositions (Example 1.1);
  - adverbial clauses (Example 5.15).

3.3 System Optimization

This section investigates the performance of STASEL as an interactive system. The functional assessment of its stylistic processing capabilities, which have little impact on response time, has been left to the next chapter. In fact, the performance inhibitors of the system lie almost entirely in the parsing process. STASEL's parser has been designed to circumvent some of the inherent problems associated with the use of a backtracking parsing mechanism, yet there are still significant fluctuations in response time according to the nature and structure of the input sentence.

As Farghaly commented, "a parser is as good as the grammar it incorporates" [Far89a, p.243], and since there is no such thing yet as a complete grammar of English, a natural language parser is bound to be restricted to a subset of possible utterances. In STASEL, this limitation has a double impact on the system's performance, first regarding functionality, and second, concerning response time.

Functionally, STASEL is not yet equipped to recognize such complicated structures as

- sentence fragments;
- interrogative sentences;
- imperative sentences;
- multiple levels of coordinated sentences;
- multiple levels of subordinate clauses;
- complex/compound sentences.

If a sentence with unrecognized structure is entered, the parser attempts to find a valid parse regardless, and applies each possible rule in turn, backtracking to the previous one whenever a rule fails. Since a phrase structure rule consists of several levels of sub-rules, the backtracking process seems to go on forever[9], particularly if the unrecognized portion is at the end of the sentence. In this sense, STASEL's parser is not robust. Heuristics capable of handling such situations need to be incorporated into the parser before any practical tests are performed in a classroom environment.

Another performance factor induced by the backtracking nature of the implementation emerges with the use of heuristics to resolve structural ambiguities. STASEL's parser attempts to resolve ambiguities without the help of pre-processing or probabilistic clues. Every disambiguation procedure is performed internally according to the linguistic information contained in the input. The parser specifically looks for the presence or absence of commas and conjunctions to determine which rules may be applied, thus reducing the amount of potential backtracking during parsing. For instance, before applying the rule for compound structures, STASEL first checks that at least one conjunction is present in the input. If this is not the case, the rule is simply bypassed. Similarly, a sentence interruption requires that the interrupting element be "framed" by two commas. The rule for interrupting sentences is not applied if less than two commas are detected in the input.

The phrase structure rules of the parser are ordered so that the rules with the most restricted entry check are examined first. If the entry checks have the same weight, the rule that takes precedence is the one that is most likely to find out the soonest that it does not apply[10].

Rule 1: interrupted sentences — two commas required;

Rule 2: compound sentences — at least one conjunction required;

Rule 3: sentences with final modifying element — at least one comma;

Rule 4: simple sentences — no requirement;

Rule 5: complex sentences — no requirement or one comma.

In addition to the initial check performed to detect spelling errors and unrecognized strings before any rule is applied, this technique considerably speeds up the parsing process, reducing the response time to a few seconds for most types of inputs, but it is of little help in ambiguous cases. Consider Example B.7:[11]

(B.7) I like skating, dancing and swimming.

[9] During tests, the parsing process had to be terminated manually when a unrecognized structure was entered. If the system had been left to run, it would have eventually found that none of its rules applied to the current input, but there is no indication of how long it would have taken to reach this conclusion.

[10] An exception to this ordering scheme is that the rule for simple sentences precedes the rule for complex sentences. This was done to allow three-way conjunction of noun phrases in the subject position of simple sentences, which would otherwise be wrongly read as a complex sentence with an initial modifying component when the first noun phrase of the conjunction is an infinitive or a gerundival phrase.

[11] Throughout this thesis, example sentences for which a system output has been produced are shown with a label heading such as (5.13) or (B.7). This label indicates where, in the thesis, the STASEL output message corresponding to the sentence is located. The first digit of the label refers to the chapter or appendix where the output can be found.

This sentence is a simple sentence with a three-way conjunction of noun phrases in the complement position. However, because of the presence of a comma *and* a conjunction, the parser will attempt to fulfil the requirements for rules 1, 2, and 3 before it enters the proper rule. Being a short sentence, Example B.7 will cause an increase in response time of the order of 5 to 10 seconds over a typical 1 to 2 seconds response time in a non-ambiguous case, an increase which is still acceptable in an interactive environment.[12]

The problem becomes more important in cases where the ambiguity lies at the end of a long sentence such as in Example 5.13:

> (5.13) To judge from the canvases on the walls of the gallery, those who are responsible are myopic.

To judge from the canvases on the walls of the gallery can be the infinitive subject of a sentence, and is recognized as such by the rule for simple sentences. However, expecting a verb to follow the infinitive subject, the rule stops when it encounters the comma after *gallery*, and the system starts backtracking. It backtracks through the entire infinitive clause before it realizes that the rule for simple sentences does not apply. The parser then enters the rule for complex sentences and re-analyzes the infinitive clause "from scratch" before finally completing the parse. The increase in response time, in such cases, is significant and can reach several minutes.

Nonetheless, the parser still preserves the communicative objective of ICALI systems by producing most responses within 1 to 5 seconds. In more than 100 example sentences tested, only a few parses (less than 10%) required more than 5 seconds of processing and in each case, the increase was caused by an ambiguity factor involving one of the following structures:

- initial modifying element of a complex sentence that can also be read as a subject noun phrase (Example 5.13);
- verb phrases in the past tense that can be read as participial modifiers of a noun (Example B.10);
- a conjunction of noun phrases that can complicate the analysis of a compound sentence (Example B.1).

Yet all examples tested resulted in a correct parse[13], an essential requirement for the stylistic processing to follow, and a consideration in language instruction that has priority over system performance. STASEL's parser would benefit from having additional disambiguation mechanisms to reduce the parsing time of ambiguous structures, but this represents a research topic of its own and diverges from the main interest of this work, the stylistic analysis of sentences.

[12] The times given in this section are an indication of the typical response I have experienced when using the system. Informally recorded, these times are solely meant to give an appreciation of the effect on response time of parsing complex and ambiguous structures. The times given here are system dependent and may thus vary depending on the external system load during parsing, the speed of the processor on which the system runs, and the version of the PROLOG compiler used.

[13] This work does not claim that the parses generated are accurate from all linguistic points of view, but that they are "correct" with regards to the requirements of this application.

Chapter 4

Syntactic Style Analysis

> *All of us who are committed to excellence in prose share a common end: a style that communicates effectively, even elegantly. And that, by and large, is a style that is clear, precise and concise.*—Joseph M. Williams [Wil81, p.183].

Clarity in sentence construction, precision in the choice of words, and conciseness in structure are all essential features contributing to the stylistic effectiveness of a sentence. These features also form the basis of the stylistic instruction provided by STASEL. In the next two chapters, we describe the implementation and operation of the stylistic analyzer modules of STASEL: the syntactic style analyzer (SSA) and the goal-directed style analyzer (GSA). These stylistic modules, though discussed in separate chapters, are both considered with regards to the stylistic principles on which they are based and the instructional feedback they provide to the student. The design features of the modules are illustrated throughout the discussion with examples of system outputs. To focus on particular system functions, as well as to save space, several of the example outputs have been edited to show only the output segment relevant to the discussion. Other output examples that are referenced, but not printed in the text, can be found in Appendix B.

4.1 Syntactic Style Analyzer (SSA)

STASEL's SSA module provides stylistic instruction at the word and sentence level by comparing the input against recognized principles of syntactic style that contribute to stylistic effectiveness in sentence writing. The instruction provided by the SSA module is geared toward two objectives: writing with precision and conciseness in a *formal* setting[1].

Unlike other style-checking systems, STASEL does not attempt to compute the many quantifiable parameters of a sentence[2], as these parameters are statistics that are more helpful in the analysis of larger texts than in determining the stylistic effectiveness of single sentences. Instead, with the SSA module, the system detects and analyzes a set of *problem features*, such as inappropriate usage choices (clichés, jargon, slang), wordiness contributors, misused phrases, and informal constructs that disrupt the structure and lessen the stylistic impact of the sentence. These problem features are not grammatical errors, but they

[1] The requirements for sentence *clarity* are discussed with the GSA module in Chapter 5.

[2] Quantifiable parameters of a sentence include: the average length of the sentence, the frequency of word occurrences, the functional type (statement, question, command, exclamation), the type and frequency of sentence openers, the amount of embedding, etc.

interfere with the writer's ability to communicate with the reader. Avoiding them, that is, respecting the principles of standard good style (as documented in style textbooks), results in "writing that is not only technically correct, but also stylistically appropriate" [NG88, p.217].

As input, the SSA module uses the stylistic parse tree built during the parsing process described in Section 3.1.2. From the tree, the module generates a high-level structural analysis of the sentence, and produces a list of stylistic flags for each word choice or sentence construction that violates conciseness requirements and formal writing principles. This higher-level structural analysis is passed to the goal-directed style analyzer (GSA) for further analysis and is displayed in the final output message to the student. Neither the structural analysis nor the list of stylistic flags are printed directly in the message; they are first processed by the information module to produce a "readable" analysis for the former, and a sequence of informative replies for the latter. Through these replies, STASEL is able to inform the student of usage and wordiness problems, point out potential structural violations, issue warning and appraisal messages, and provide remedial feedback.

4.2 Structural Analysis

The information contained in the structural analysis produced by the SSA module is fundamental to the design of STASEL, because it provides the basis for the clarity analysis that follows in the GSA module. The structural analysis serves two purposes: it is used internally as input to the GSA module for the goal-directed analysis, and it is displayed in the final output message, along with the result of the SSA and GSA analyses, to help the student understand the structure of the sentence and better appreciate the content of the stylistic messages. The structural analysis output used internally contains various mnemonics and abbreviations that would make little sense to a student. Consequently, before being displayed, the analysis is "translated" into intelligible English with the help of the information module, so that each relevant feature is associated with a brief description that appears in the output instead of the mnemonic.

As an example, consider this sentence:

(4.1) Despite the weather and the road conditions, I returned safely.

The internal analysis used by the GSA module is shown in Figure 4.1 while the corresponding output analysis, edited by the information module, appears in the last segment of Example 4.1.

```
imc(modC=
        prepositional(pp2),

    clause(
          subj(simpleNP(none)),
          noComp,
          mod(adverb)))
```

Figure 4.1 Structural analysis of Example 4.1

```
-----------------------------------------------------------------
ANALYSIS OF:  Despite the weather and the road conditions, I
              returned safely.

-----------------------------------------------------------------

 <=> GOOD: parallel construct achieved

-----------------------------------------------------------------

 <=> SENTENCE IS CLEAR: resolution

-----------------------------------------------------------------

 <=> STRUCTURE: initial modifying component + clause

    initial modC -->
        pp that contains a 2-way conjunct np

    clause -->
        subj: simple noun phrase (np)
        mod: adverb
-----------------------------------------------------------------
```

Example 4.1

The structural analysis not only gives a high-level description of the sentence's constituents, but also judges the complexity of the structures encountered. In the analysis, elements of a sentence are described as functional and grammatical entities can either be termed *simple* or *complex* according to the structure they display. Each constituent of the sentence is thus:

- described according to its function within the sentence (subject, complement, modifier, interrupting element, modifying element) that, in turn, is expressed as one or more grammatical entities (noun phrase, prepositional phrase, adverb, relative clause, etc.);
- identified as being either structurally *simple* or *complex*.[3]

The structural analysis is built concurrently with the list of stylistic flags during the SSA's examination of the input stylistic parse tree. To illustrate this process, consider a sentence that displays some structural complexity:

(4.2) John bought books about computers and music.

This sentence is structurally complex because it is not clear whether John bought books on computers and books on music, or whether he bought books on computers and then bought some music items (scores, tapes, or records). In other words, without contextual information, it is not possible to determine whether *computers and music* or simply *computers* is the object of the preposition *about*. If the complex element, in this case the prepositional phrase, is placed at the end of the sentence, the ambiguity no longer remains:

(B.9) John bought music and books about computers.

[3] In the output version of the structural analysis, the label `complex >>` appears in the description whenever a constituent is found to be complex.

The stylistic parse tree of Example 4.2 is shown in Figure 4.2. It consists of

1. two top-level "style" frames: the subject frame `style(proper,[])` and the verb frame `style(active,AttList)`, where `AttList` is the list of attributes of the frame;
2. one second-level style frame that describes the two-way conjunction of noun phrases (conjNP2) that form the complement;
3. two np style frames on the third level that describe the structure of the complement in detail;
4. one pp style frame on the fourth level that represents the prepositional phrase introduced by *about*;
5. a np-frame on the fifth level that describes the object of the prepositional phrase.

```
SUBJECT:
   style(proper,[])

VERB PHRASE:
   style(active,[
                comp=style(conjNP2,[
                                   style(np,[
                                            style(pp,[
                                                     pp=prep(about)
                                                     style(np,[])])])
                                   style(np,[study=music])])])
```

Figure 4.2 Stylistic parse tree of Example 4.2

The SSA module analyzes each style frame by applying a series rules designed to detect stylistic problems to each of the attributes of the frame. Whenever a rule succeeds in finding a problem, a flag is inserted into the output list of stylistic flags. The SSA module first examines the top-level frames and then proceeds in a recursive fashion until all levels have been examined. Thus, the first style frame that produces an output, consisting of an analysis and a list of flags is the fifth-level np-frame. This output is then passed back to the fourth-level pp-frame, the fifth-level list of flags is appended to that of the fourth-level, and a new analysis is generated. The recursive process continues until each frame has been processed. Carefully encoded *conversion heuristics* are used to translate the content of a sub-level analysis into a higher-level analysis. The complete recursion necessary to process the sentence of Example 4.2 goes as follows:

- 5TH LEVEL:
  The frame structure `style(np,[])` is converted to simple noun phrase.
  System mnemonic: `simpleNP(none)`
  Partial flag list: `[]`
  Corresponding element: *computers*
- 4TH LEVEL:
  The pp-frame structure receives the output of the fifth level and converts the simple

noun phrase analysis to a simple prepositional phrase analysis.
System mnemonic: pp
Partial flag list: []
Corresponding element: *about computers*

- 3RD LEVEL:

  - The first np-frame of the conjNP2 structure receives the fourth-level output and converts it to a simple noun phrase containing a simple prepositional phrase.
    System mnemonic: `simpleNP(pp)`
    Partial flag list: []
    Corresponding element: *books about computers*
  - The second np-frame of the conjNP2 is converted to a simple noun phrase.
    System mnemonic: `simpleNP(none)`
    Partial flag list: []
    Corresponding element: *music*

- 2ND LEVEL:
  The rule that checks for ambiguities and lack of parallelism[4] in the organization of a compound structure detects a complexity in the information received from the third level, and the `conjNP2`-frame is converted to a complex two-way conjunction of noun phrases. A faulty parallelism flag (nonP) is added to the list of stylistic flags.
  System mnemonic: `complexNP(np2)`
  Partial flag list: `[nonP]`
  Corresponding element: *books about computers and music*

- TOP LEVEL:
  At this level, each functional entity is analyzed:[5]

  - The subject frame is analyzed and the `proper`-frame is converted to a simple noun phrase[6].
    System mnemonic: `subj(simpleNP(none))`
    Partial flag list: []
    Corresponding element: *John*
  - The verb phrase's `active`-frame receives complement information from the previous levels and converts the frame to a complex complement and no verb modification. The partial list of flags also carries on from the previous levels.
    System mnemonic: `comp(complexNP(np2))` for the complement
    System mnemonic: noMod for the verb modification
    Partial flag list: `[nonP]`
    Corresponding element: *bought books about computers and music*

[4] Parallelism is achieved by grouping items of similar functional role into a series with each item expressed in the same grammatical form, as in *I like to skate and to swim.* See Section 4.3.3 for a more detailed discussion of parallelism.

[5] If other entities besides the subject and the verb phrase are present, for instance, in a compound sentence structure or in a sentence that contains a modifying element, the top-level analysis is performed on each entity in turn.

[6] Frame statements for proper names and pronouns both produce the same analysis as a simple noun phrase that has no post-modification (that is, has no attached prepositional phrase or modifying clause).The system mnemonic for all three cases is `simpleNP(none)`, where none stands for no post-modification.

- SENTENCE LEVEL:
  This level completes the recursive process, produces a final list of flags, and builds the structural analysis from the output produced at each preceding level.
  STRUCTURAL ANALYSIS:
  `clause(subj(simpleNP(none)),comp(complexNP(np2)),noMod)`
  List of stylistic flags: [nonP]

Once the SSA module has completed the breakdown of the stylistic tree, the structural analysis is passed to the GSA module and a clarity diagnosis is produced. The final list of flags, the structural analysis, and the clarity diagnosis are then processed by the information module and displayed to the student as shown in the output of Example 4.2.

```
-----------------------------------------------------------------

ANALYSIS OF:  John bought books about computers and music.

-----------------------------------------------------------------

  *  faulty parallelism

-----------------------------------------------------------------

 *** sentence is unclear ***

  >> complex structure in the predicate

-----------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
         subj: simple noun phrase (np)
         comp: complex >> conjunction of 2 noun phrases

-----------------------------------------------------------------
```

Example 4.2

4.3 List of Stylistic Flags

Good response handling demands far more than merely showing right answers. It must direct student attention to errors and offer assistance in understanding and rectifying them.—Jack Burston [Bur88]

Burston's recommendation was taken into consideration when laying out the output segment associated with the list of stylistic flags produced by the SSA module, for this segment carries an important portion of the instructional response provided by STASEL. When processed by the information module, each flag is associated with a detailed explanation that contains a description of the cause of the problem and a remedial statement to help the student correct it. Each explanation is printed in the output message with a heading that identifies the type of stylistic problem encountered. STASEL employs five heading types to differentiate between categories of stylistic flags:

- `USAGE-->`
  indicates a diction problem, and may refer to a lexical choice that is inappropriate in formal writing, to a word that is incorrectly used, or to an expression that has been devalued by overuse;

- `[W]`
  signals a lack of conciseness, detecting the presence of wordy elements, redundant forms, unnecessary constructions, and meaningless structures;

- `*`
  denotes structural constructions that may be potentially damaging to the style of the sentence, including excessive noun modification, split infinitives, faulty parallelism, and the use of the passive voice;

- `WARNING:`
  handles special cases that do not fall under the above three categories, such as the use of intensifiers, double negations, vagueness, and unusual syntactic constructions;

- `<=> GOOD`
  appears whenever the SSA fails to detect stylistic problems in the input sentence, or if a particularly noteworthy sentence construction pattern, such as parallelism, has been found.

The following sections provide a detailed outline of the various stylistic problems (and some good points) detected by STASEL. As shown in Section 4.2, each problem is summarized in a stylistic flag that, in turns, corresponds to at least one stylistic rule of the recursive examination process performed by the SSA module. In total, the SSA module recognizes 81 different stylistic flags, which translate in the code to 22 sentence-level rules, 135 stylistic rules used in the recursive process, and 88 conversion heuristics that produce 74 different complexity analyses.

4.3.1 Usage Considerations

Usage designates the conventions that determine *how* language is used in certain situations. These conventions can be expressed as precise lexical features, characteristic of the level of discourse pursued (in our case, formal writing). Usage conventions are less restrictive than grammar rules, but they should be respected, as most of them contribute to clarity and economy of expression in writing.

There are many levels of usage in written English (colloquial, general, formal, technical, etc.), each with their own characteristic features. For instance, Norton and Green [NG88] associate the following practical usage parameters with formal writing:

- *Vocabulary:* no contractions, no colloquialisms, no specialized words unless writing for a specific audience, a mixture of abstract and concrete words;

- *Sentence structure:* complete sentences, no fragments, varying length but with predominance of long and complex sentences.

STASEL recognizes its own usage parameters of vocabulary choices, but leaves sentence structures to the *-type flags. The following usage categories are flagged:

Slang terms: A word or expression that is extremely informal and has the connotation of particular social situations. Slang terms should be avoided not only because they are usually inelegant, but also because their meaning is restricted and often vague.

(4.3) It is kind of hard to be a *cop* in this society.

Colloquial terms: A word or expression that is appropriate to a conversational level of usage, but sounds out of place in formal writing.

(B.8) Dancing has a *super* effect on me.
(B.23) The computer power energy need of this *lab* is tremendous.

Jargon: Misused technical term or expression. Technical language, such as the highly specialized vocabulary used in a particular field, should be restricted to expert writing.

(B.29) The engineers still have to get a *buffer* but they are about done.

Taboo expressions: Words that are not acceptable in polite use. (B.16)

Archaic words A word or expression that is no longer in common use but may be found in literary works or used to impart a historical colour to contemporary writing.

(B.20) *Thou shouldst* give me a purple-coloured stamp.

Clichés: A word or expression that has lost much of its force through overexposure. Clichés are dull and unoriginal, and often add unnecessary weight to the sentence.

(B.25) I left the scene of the accident, shocked, dizzy, and *white as snow.*
(4.4) Hopefully, we have reached *the moment of truth.*

Pompous diction: A pompous (or pretentious) word is a fancy one used where a common term can be as efficient. Using simpler words results in writing clearer sentences.

(B.3) The artist provides a dreamy background done in yellow and *bistre* brushstrokes.[7]

For single word flagging, the usage category is explicitly listed as an attribute of the lexicon entry, such as `collo=lab`, while for longer expressions, the usage category is associated with the entire expression, as in `cliche=mot` where `mot` stands for *moment of truth*. When an inappropriate usage attribute is detected in the stylistic parse tree, the SSA module simply transfers the given attribute to the list of stylistic flags. The attribute is eventually displayed in the ouptut message with the heading `USAGE -->`, as shown in Example 4.3.

The SSA module also detects usage problems associated with using incorrect grammatical forms of a word. A common example of such misuse is to put an adjective or a preposition in the place of an adverb as in these sentences:

(Ex. B.30) It felt like my world had collapsed and it was *just* terrible.

(Ex. B.29) The engineers still have to get a buffer but they are *about* done.

Misused words may be acceptable in conversation, but not in formal writing. For instance, the use of *kind of* to mean "quite", as in Example 4.3, is highly informal and should be avoided. Such errors are anticipated in the lexicon by coding the wrong grammatical form

[7] This example is taken from DiMarco's thesis [DiM89]. Other examples taken from the same source are 5.3, 5.6, 5.9, 5.10, 5.13, 5.15, 5.16, 5.18, B.1, B.2, B.3, B.4, and B.6.

```
-----------------------------------------------------------------

ANALYSIS OF:  It is kind of hard to be a cop in this society.

-----------------------------------------------------------------

 [W] anticipatory construct introduced by "it"

  USAGE--> "kind of" is colloquial when used as an adverb

  USAGE--> slang word "cop"

-----------------------------------------------------------------

 <=> SENTENCE IS CLEAR: monoschematic

-----------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
         subj: simple noun phrase (np)
         comp: adjectival phrase
         comp: infinitive clause

-----------------------------------------------------------------
```

Example 4.3

as an entry and adding a special attribute (**special=kind-of**) to the frame list. This attribute is then directly transferred to the list of flags for display in the output message.

A last category of words, widely recognized as being persistently overused or misused in student prose, are flagged in the lexicon by the **special=** attribute and correspond to a unique remedial message. These include the verb *to seem* (Ex. B.5), the adverbs *actually* (Ex. B.14) and *definitely* (Ex. B.12), and the infamous *hopefully* (shown with its remedial message in Ex. 4.4).

```
-----------------------------------------------------------------

ANALYSIS OF:  Hopefully, we have reached the moment of truth.

-----------------------------------------------------------------

  USAGE--> "hopefully" in the sense of -I hope- or -it is hoped-
           is overused and informal. It is better to restrict the
           sense to "in a hopeful way"

  USAGE--> "the moment of truth" is a cliche

-----------------------------------------------------------------
```

Example 4.4

4.3.2 Avoiding Wordiness

All writing should be as concise as it can be.—Sarah Norton [NG88, p.224]

One of the most frequent complaints of teachers is "the general sin of wordiness". As Sheridan Baker puts it, "Verbosity is a disease with the following symptoms: severe inflation of the language, difficulty in following the point, extreme drowsiness" [Bak73, p.88].

Baker is not the only one to advocate conciseness as an essential feature of good style. Most textbooks on style ([RR65, Kan83, Eas70, Wil81]) recognize the importance of avoiding wordy structures, and each proposes a set of guidelines to rectify the situation. The following list gives a summary of the most common sources of wordiness and shows how STASEL handles the problem. The system flags the structures that contribute to wordiness in a sentence with the `[W]` heading, as shown repeatedly in the output of a particularly wordy and badly constructed sentence presented in Example 4.5.

```
-----------------------------------------------------------------

ANALYSIS OF:  In my personal opinion, it should be known that the
              problem of computers is not well understood.

-----------------------------------------------------------------

  *  2 passives detected

 [W] wordy construct around "opinion"
     In good writing, adjectives and adverbs should link as
     directly as possible to what they really modify.
     A better link is to replace the prepositional phrase with
     an adverb derived from the adjective modifying "opinion".

 [W] "personal" redundantly modifies "opinion"
     It is a pointless repetition of the same idea

 [W] anticipatory construct introduced by "it"

 [W] the "problem of" is a weak construct which says
     nothing. It is better avoided

 WARNING: use of attitude verbs such as "know" is unnecessary

 WARNING: use negative ("not") only if necessary for emphasis or
          contrast. Otherwise, use a positive form

-----------------------------------------------------------------
```

Example 4.5

- *Redundant pairs*

  The combination of a simple word with a modifier that includes an aspect of the word implied by definition constitutes a redundant pair. For example, there is no need to write "*final* conclusion" for a conclusion is final by definition. To detect this problem, STASEL stores a list of common redundant pairs and adds a lexicon attribute `red=` to each potential member of a pair. In the SSA module, a stylistic rule detects the presence of two `red=`, either consecutively in the sentence or separated by a preposition, and compares them to the list of possible pairs. If the comparison is successful, a flag is added to the list of stylistic flags.

Redundant pairs are identified in the output message as shown in Ex. 4.5 and 4.6.

```
------------------------------------------------------------------

ANALYSIS OF:  The process of growth is clearly evident after a
              period of time.

------------------------------------------------------------------

 [W] the "process of" is a weak construct which says
     nothing. It is better avoided

 [W] "clearly" redundantly modifies "evident"
     It is a pointless repetition of the same idea

 [W] restrict the use of vacuous words such as "clearly"

 [W] "time" redundantly modifies "period"
     It is a pointless repetition of the same idea

------------------------------------------------------------------
```

Example 4.6

- *Redundant categories*
  Redundant categories occur when a plain word is coupled with a needless noun that implies its category.

  (B.20) Thou shouldst give me a *purple coloured* stamp.
  (B.21) My brother says that a computer keyboard is *rectangular* in *shape*.

  In STASEL, potential members of a redundant category are identified in the lexicon with the use of two attributes: `cat=X` to identify the category, and `X=word` to identify the word that may be redundantly categorized by category X. Whenever the SSA module detects a `cat=X` attribute, it looks for the presence of a corresponding `X=word` flag where X is the same for both attributes. If it succeeds, a flag is added to the list of stylistic flags.

- *Wordy constructs*

  - If a long expression is used where an equivalent word would suffice, as when writing *despite the fact that* (Ex. B.13) instead of *although* or *even though*, the result is unnecessarily wordy. In STASEL, such expressions are listed in the lexicon and recognized during the parsing process.
  - A relative clause introduced by a relative pronoun followed by an auxiliary form of *to be* constitutes a wordy passive construct.

    (4.7) The book *that was* bought by John is written on paper *that was* recycled.

  Style books suggest dropping the combination of pronoun and *be* whenever possible, and linking the phrase with the remaining participle. STASEL recognizes wordy passive constructs during the parsing of relative clause and adds an attribute to the frame of the clause for later processing by the SSA module. Example 4.7 demonstrates how wordy passive constructs affect the stylistic quality of a sentence by showing two contrasting analyses performed by STASEL.

```
        ----------------------------------------------------------------
>>>>    ANALYSIS OF:  The book that was bought by John is written on
                      paper that was recycled.
        ----------------------------------------------------------------
          *  3 passives detected

         [W] wordy passive construct in relative clause
             Adjectival wordiness occurs when the writer uses a relative
             clause to introduce a participle that could be attached to
             the noun directly. Occasionally, clarity and emphasis justify
             writing out the entire clause, but in most cases, it is
             better to simply drop the relative pronoun and be auxiliary.
        ----------------------------------------------------------------
         <=> SENTENCE IS CLEAR: monoschematic

               But it would be clearer without the
               excessive number of passive verbs
        ----------------------------------------------------------------
                                   *
                                   *
        ----------------------------------------------------------------
>>>>    ANALYSIS OF:  The book bought by John is written on recycled
                      paper.
        ----------------------------------------------------------------
          *  2 passives detected

         <=> GOOD: stylistically correct
        ----------------------------------------------------------------
         <=> SENTENCE IS CLEAR: monoschematic
        ----------------------------------------------------------------
```

Example 4.7

- Wordy constructs also arise when a vacuous (weak) word is used to link adjectives, adverbs, or relative clauses to the rest of the sentence. Vacuous words, such as *way, kind, type*, have a proper English sense, but they generally carry little meaning and are frequently inserted to fill a grammatical gap or to make the sentence appear to say something more significant than it really does. The three shortened analyses presented in Example 4.8 show that STASEL treats wordy constructs caused by the presence of vacuous words very carefully and becomes satisfied only when the vacuous item has been removed.

- *Anticipatory constructs*
  An anticipatory construct is created when a pronoun that has no meaning of its own is used to fill the subject position. Such constructions are usually introduced by one of the pronouns *it, this, these, there*, followed by a verb such as *to be* or *to seem* that is used to link the pronoun to the main idea of the sentence (See Example 4.9).

  Anticipatory constructions use more words than comparable direct statements. Sometimes, the extra words serve a valid purpose, for emphasis or idiom, but when they do not, the writer should recast the sentence to make better use of its main elements.

```
      ----------------------------------------------------------------
>>>>  ANALYSIS OF:  He skated in an elegant way.
      ----------------------------------------------------------------
       [W] wordy construct around vacuous word "way"
           In good writing, adjectives and adverbs should link as
           directly as possible to what they really modify.
           A better link is to replace the prepositional phrase with
           an adverb derived from the adjective modifying "way".
      ----------------------------------------------------------------
                                  *
                                  *
      ----------------------------------------------------------------
>>>>  ANALYSIS OF:  He skated in a way that seemed elegant.
      ----------------------------------------------------------------
       [W] wordy construct around vacuous word "way"
           In good writing, adjectives and adverbs should link as
           directly as possible to what they really modify.
           A better link is to replace "way" and the
           relative clause following it with an appropriate adverb.
      ----------------------------------------------------------------
                                  *
                                  *
      ----------------------------------------------------------------
>>>>  ANALYSIS OF:  He skated elegantly.
      ----------------------------------------------------------------
       <=> GOOD: stylistically correct
      ----------------------------------------------------------------
```

Example 4.8

```
----------------------------------------------------------------

ANALYSIS OF:  There is an engineer working on the problem.

----------------------------------------------------------------

 [W]  anticipatory construct introduced by "there"

----------------------------------------------------------------
```

Example 4.9

For example, instead of *There is X*, the writer should introduce the sentence with *X*, *Some X*, or *Many X*, unless the existence of X is less important than its properties.

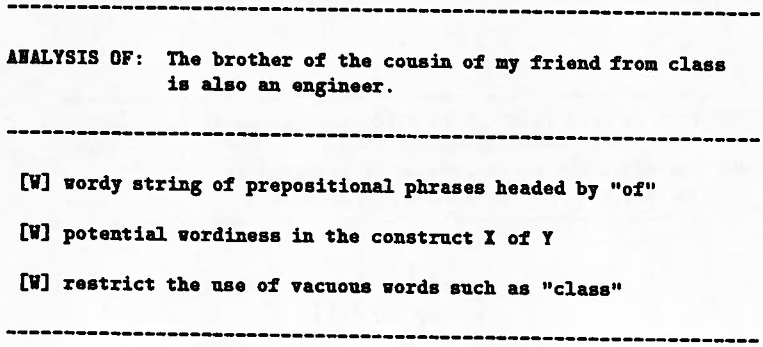
If one of the introductory pronouns *it, this, these, there* is detected in the subject position during parsing, the parser adds a special ant attribute to the stylistic parse tree. When this attribute is detected by the SSA module, the rule for anticipatory constructs is invoked. The rule checks for the presence of a following *be* or *seem* verb, and produces a flag if it finds it.

- *Abstract nouns*
  Nouns that are based on, or derived from, verbs and adjectives, such as *process, factor, concept, subject, thought, field*, should be used with caution. If one consistently expresses actions in abstract nouns followed by weak verbs (*have, be, do*), the chance is high that the prose will be both heavy and unclear. Abstract nouns are not inherently wrong, but they can often be eliminated by either rewriting the clause into a participial phrase (see contrast between Ex. B.27 and B.28), or dropping the abstract noun altogether.

  In the field of medicine → In medicine

  In STASEL, abstract nouns are identified as such in the lexicon. When an abstract noun is detected, a rule of the SSA module is invoked to determine whether the noun is part of a wordy construct by detecting the presence or absence of a weak verb in the verb phrase frame.

Other well-known wordiness contributors detected by STASEL include vacuous words (Exs. B.11, B.17, 4.6), strings of prepositional phrases headed by *of* (Ex. 4.10), repeated subjects in compound sentences (Ex. B.15), and wordy constructs around weak verbs (Exs. B.10, B.17).



```
-----------------------------------------------------------------

ANALYSIS OF:  The brother of the cousin of my friend from class
              is also an engineer.

-----------------------------------------------------------------

 [W] wordy string of prepositional phrases headed by "of"

 [W] potential wordiness in the construct X of Y

 [W] restrict the use of vacuous words such as "class"

-----------------------------------------------------------------
```

Example 4.10

4.3.3 Structural Considerations

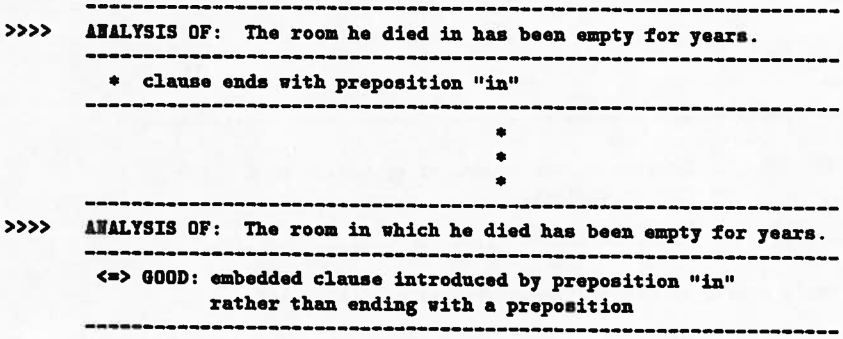
Stylistic messages headed by * point out structural constructions that may potentially hinder the flow of a sentence and keep the writer from linking one idea to another clearly and explicitly. The SSA module recognizes five potentially damaging structural constructions and provides principled guidelines to help the student in either removing or correcting the inappropriate structure. These principles do not limit the student to only one solution. They simply suggests means of conveying and stressing ideas in a sentence by carefully choosing the sentence structure.

1. *Excess modification*

   - with nouns (Ex. B.23);
   - with adjectives (Ex. B.25);
   - with adverbs (Ex. B.11);

   Sentences can grow both long and confusing when too many modifiers are used. Qualifying phrases or strings of modifiers necessarily break up the grammatical flow, and may even fail in their presumed purpose of careful definition. A series of strong words, such as adjectives, nouns, and adverbs, may also leave the reader wondering just which one is supposed to be the most important, so that much of the modifying effect is lost. For these reasons, multiple modifiers of a single idea should be used with caution.[8]

2. *Clause ending with a preposition* (Ex. 4.11)
   Clauses that end with a preposition that is not closely related to the verb it modifies (that is, a preposition that does not act as a verb particle as in *look up*) carry an informal connotation. In formal writing, it is best to recast the sentence with the informal clause introduced (rather than ended) by the preposition. This principle is illustrated by two contrasting analyses in Example 4.11.



```
        ------------------------------------------------------------
>>>>    ANALYSIS OF:  The room he died in has been empty for years.
        ------------------------------------------------------------
          *  clause ends with preposition "in"
        ------------------------------------------------------------
                                     *
                                     *
                                     *
        ------------------------------------------------------------
>>>>    ANALYSIS OF:  The room in which he died has been empty for years.
        ------------------------------------------------------------
          <=> GOOD: embedded clause introduced by preposition "in"
                    rather than ending with a preposition
        ------------------------------------------------------------
```

Example 4.11

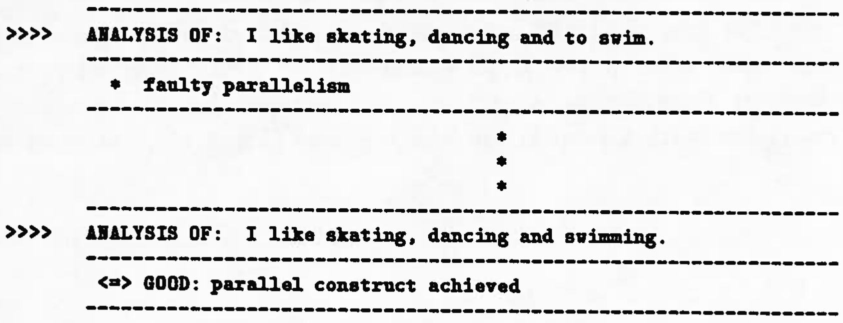
[8] Another particularly important source of confusion caused by the use of modifiers, but not treated in STASEL because of lack of semantic processing capabilities, is the *dangling* modifier problem. A modifier is said to be dangling if there is no appropriate word or phrase in the sentence to which the modifier can be associated, as in:

> Walking in the park, a squirrel nearly bit me.

This example is confusing because the initial modifying clause does not apply to the subject (*squirrel*) but to the person walking in the park (*me*), and thus, should be rewritten.

3. *Split infinitives* (Ex. B.22)
   An infinitive verb is said to have been split if a word or a group of words, usually adverbs, have been inserted between the infinitive marker *to* and the verb root. Although occasional short interruptions are allowed and even provide pleasant variety in style, the grammatically related parts of an infinitive should not be separated unduly. Split infinitives can become confusing and should be avoided when other arrangements, such as putting the splitting element at the end of the infinitive phrase, will serve equally well.

4. *Faulty parallelism* (Ex. 4.2, 4.12)
   Parallelism is characteristic of clear and economical thinking. In writing, parallel structures offer a clear and flexible way of ordering ideas in a sentence. A parallel structure is achieved by grouping items of similar importance into a series where each item is expressed in the same grammatical construction. A common failure to observe parallelism occurs when one of the elements of a series does not exhibit the same grammatical form, as shown in Example 4.12.



```
    -----------------------------------------------------------------
>>>> ANALYSIS OF:  I like skating, dancing and to swim.
    -----------------------------------------------------------------
      * faulty parallelism
    -----------------------------------------------------------------
                                 *
                                 *
                                 *
    -----------------------------------------------------------------
>>>> ANALYSIS OF:  I like skating, dancing and swimming.
    -----------------------------------------------------------------
     <=> GOOD: parallel construct achieved
    -----------------------------------------------------------------
```

Example 4.12

5. *Excessive use of passive verbs* (Ex. 4.7)
   A writer should use active verbs as often as possible. Style can be made more direct, vigorous and concise if the writer avoids unnecessary passive verbs. There are occasions when passives are legitimate, but if used in excess, they slow down the rhythm of the sentence. Passives are useful in some occasions to

   - help keep the focus of the sentence on someone or something important that is acted upon;
   - describe an action when the writer does not know or does not care to say who or what performs it;
   - put the subject of the action at the end of the clause where the writer can easily attach a long modifier.

4.3.4 Warning and Appraisal Messages

Warning messages are meant to inform the student of the presence of a problem in the sentence that is not covered under the previous flag categories but may contribute to poor style. STASEL issues a message, identified with the heading `WARNING:`, whenever one of the following stylistic guidelines[9] is violated:

- Avoid using vague words such as *thing, nice, people* (Exs. B.2, 5.17, B.17);
- Limit the use of attitude verbs such as *to know, to feel, to believe, to think* (Ex. 4.5);
- Do not overuse common intensifiers such as *very, terribly, extremely* (Ex. B.17);
- Be careful that a referent exists when using a demonstrative pronoun to introduce a main clause (Exs. 5.13, B.10);
- Limit the use of double negations such as *not impossible* (Ex. B.22);
- Use negative terms only if necessary for emphasis or contrast (Exs. 4.5, B.11);
- Do not start sentences with a conjunction such as *like, and, but* (Ex. B.19);
- Limit the use of weak verbs such as *have, be, do* (Ex. B.29).

Instruction cannot be complete without some appraisal and encouragement. STASEL recognizes this need and issues a `<=> GOOD` statement for stylistically correct and clear sentences, and whenever a noteworthy structure, such as a parallel construction (Ex. B.7) or a clause that could end with a preposition but does not (Ex. 4.11), has been detected.

[9] The stylistic guidelines used in warning messages were compiled from various textbooks on style ([Wil81, RR65, Bak73, NG88, Eas70]).

Chapter 5

Goal-Directed Style Analysis

An important characteristic of STASEL, which also distinguishes the system from all other style-analyzing programs, is its ability to judge sentences according to precise stylistic goals. The present implementation concentrates on the single goal of *structural clarity*, yet the system has been designed so that the inherent modularity of the goal-directed process will allow the analysis of other goals such as emphasis, concreteness, and formality to be easily implemented as future refinements of the system.

In STASEL, the clarity analysis of sentences is handled by the goal-directed style analyzer (GSA) module. The *clarity rules* contained in the module are based on formal principles of goal-directed stylistics that enable the system to perform a clarity analysis that not only determines whether a given sentence is structurally clear, but also instructs the student in the reasons that led to the diagnosis.

As input, the GSA module uses the high-level structural analysis produced by the SSA module. As seen in Section 4.2, this structural analysis includes a precise description of the sentence and a complexity analysis of each sentence component that facilitates the coding of goal-directed stylistic rules. Moreover, since the structural analysis is general in form and independent from the GSA module's operation, it can readily be used to analyze other stylistic goals. In fact, the analysis of additional goals would require no modification to the existing code other than the addition of stylistic rules specific to each new goal.

5.1 Theoretical Foundation

The stylistic principles used to judge the structural clarity of sentences in STASEL are based on the theoretical model of goal-directed stylistics developed by DiMarco for the purpose of machine translation [DiM89]. In order to formalize the notion of "stylistic features" in sentences, DiMarco defined a vocabulary of *abstract* elements of style that describe the stylistic effects of both individual sentence components and general sentence structure. With this vocabulary, DiMarco was able to provide a formal description of the syntactic constructs that characterize specific stylistic goals such as clarity. This formal description, which DiMarco provides in the form of a stylistic grammar based on the abstract elements of style defined in the vocabulary, forms the theoretical basis for the design of STASEL's GSA module.

5.1.1 The Concept of Concord and Discord

The stylistic vocabulary used to formulate the goal-directed stylistic rules for clarity is based upon the notions of *concord* and *discord*, for it is DiMarco's contention that "style is created by patterns of concord and discord giving an overall integrated arrangement" [DiM89, p.40]. She defines these notions as follows:

concord (DiMarco) A set of syntactic structures whose inter-relationships create the following stylistic effect: unity of style, agreement, accord, stability, and no requirement for resolution (to be defined in Section 5.1.2).

discord (DiMarco) A set of syntactic structures whose inter-relationships create the following stylistic effect: disunity of style, disagreement, contention, conflict, incongruity, and requirement of resolution.

DiMarco's definitions refer to abstract concepts. In STASEL, however, these concepts have been expressed in highly practical terms that correlate each concept to explicit syntactic structures. From STASEL's point of view, the notions of concord and discord have been extended to mean the following:

concord (STASEL) A syntactic element[1] is concordant if it displays clarity and stability on its own, without ambiguity of attachment or excess structure. In practice, this implies that in a sentence, any grammatical entity that is not *complex*[2] in structure and that clearly attaches to the rest of the sentence is concordant. By extension, any functional entity of a sentence is concordant if all the grammatical entities with which it is associated are concordant, and, at the top level, a sentence is concordant if all its functional entities are concordant. For example, a simple noun phrase in the subject position is concordant.

discord (STASEL) A syntactic element is discordant if it produces conflict, incongruity, or ambiguity because it is structurally complex and/or because its position in the sentence prevents a non-ambiguous attachment. In practice, any *complex* grammatical entity contributes to discord. Modifying elements that consist of a clause without a subject, a phrase displaying little structure (such as a simple adjectival or adverbial phrase), or an unusual construction (such as an inverted conditional phrase) are also discordant. For example, a prepositional phrase in the initial modifying position (that is, introducing the main clause) is discordant because it lacks an explicit subject and it may not be structurally clear which element it modifies, particularly if the following main clause contains many elements.

Discordant syntactic structures are by no means wrong; they are commonly found in written texts, often bringing variety and pleasant relief from monotonous writing. If used carelessly, however, they may prove difficult to read and contribute to poor style.

Section 5.2 provides a detailed description and examples of the concordant and discordant syntactic structures recognized by STASEL.

[1] A syntactic element, in STASEL, is any group of words within a sentence that can be defined as a grammatical or functional entity in the structural analysis of the SSA module. Noun phrases, prepositional phrases, infinitive clauses, main clauses, and modifying elements are all forms of syntactic elements.

[2] The characteristics that distinguish a *complex* syntactic structure from a *simple* one are outlined in Section 5.2.2.

5.1.2 Clarity Features Adapted from DiMarco's Model

With the notions of concord and discord, DiMarco defined a set of general rules that correlate patterns of syntactic structures with the goal of clarity [DiM89, p.105]. These rules are based on the abstract elements of style that she uses to describe the stylistic features of a sentence. In the GSA module, each element that DiMarco identifies as inducing a sense of clarity to a sentence has been refined and are made to correspond to specific sentence constructions in the form of a clarity rule. The following types of sentences, named after five of the abstract elements in DiMarco's vocabulary, but extended to suit the requirements of stylistic instruction, are associated with the goal of clarity:

1. **monoschematic**
   A monoschematic sentence consists of an independent main clause with no excess structure (i.e. a concordant main clause) and no dependent modifying elements.

   (5.3) I told him that she was wrong.

2. **centroschematic**
   A centroschematic sentence consists of a concordant main clause accompanied by a concordant modifying element that either precedes, follows, or surrounds the main clause.

   (5.8) If we can judge from the canvases on the walls, those who are responsible are myopic.

3. **resolution**
   A sentence that produces a resolution is a special type of centroschematic sentence. Such sentences consist of an discordant initial modifying element followed by a highly concordant monoschematic clause that resolves the initial incongruity. The syntactic structures associated with a highly concordant monoschematic clause are more restricted than those of a simple monoschematic sentence, to ensure a clear resolution. Nonetheless, a resolution is not as clear as a centroschematic construct because of the presence of a discordant element in the initial position (whereas the elements of a centroschematic sentence are all concordant).

   (5.2) Despite the weather and the road conditions, I returned safely.

4. **positive interruption**[3]
   A positive interruption consists of an interrupted monoschematic sentence in which the interrupting element is concordant.

   (5.15) The famous collector, when he was unable to acquire certain canvases, bought copies.

5. **compound structure**[4]
   A compound structure is a sentence that consists of the coordination of two balanced monoschematic elements (elements that display equivalent grammatical forms), and no modifying elements.

   (5.18) The style was formed and the principles were acquired.

[3] In DiMarco's model, a positive interruption is called a *counterpoise*.

[4] In DiMarco's model, a compound structure is called a *homopoise*.

DiMarco also defined a set of syntactic constructions that prevent clarity (i.e, that induce "obscurity"). She associates a lack of clarity in sentences with structural complexity, that is, sentences that contain too many dependent clauses or exhibit too much imitation (such as excessive noun modification and multiple conjunction of noun phrases). Interrupted sentences in which the interrupting element is discordant are also regarded as being unclear. Finally, DiMarco notes that the combination of coordination and clause dependency produces a discordant effect that prevents structural clarity. These "obscurity" principles were used as a basis for the design of STASEL to distinguish between simple (concordant) and complex (discordant) syntactic structures.

5.2 Clarity Rules in STASEL

The general principles outlined in Section 5.1.2 were combined and refined in STASEL to produce clarity rules. The GSA module contains seven clarity rules: each clarity feature is associated with one rule except for the centroschematic feature (two rules to distinguish between initial and final modifying elements) and the positive interruption feature (two rules to distinguish between concordant and discordant interrupting elements). The following presents a high-level description of the GSA's clarity rules in pseudo-PROLOG form[5]:

```
monoschematic :-
      concordant main clause.

centroschematic :-
      concordant initial modifying element,
      monoschematic clause.

centroschematic :-
      concordant final modifying element,
      monoschematic clause.

resolution :-
      discordant initial modifying element,
      highly concordant main clause.

positive_interruption :-
      concordant interrupting element,
      monoschematic clause.

positive_interruption :-
      discordant interrupting element,
      highly concordant main clause.

compound_structure :-
      monoschematic first clause,
      monoschematic second clause,
      balanced first and second clause.
```

[5] In PROLOG, the symbol ":-" means *if* and a comma in the body of the rule is read as *and*.

If a clarity rule fails, the GSA module enters a set of backup rules to determine the cause of the failure. This mechanism is used to generate the remedial and instructional feedback that is printed in the GSA output segment along with the clarity diagnosis. Each set of backup rules contains a default rule that applies when all other backups have also failed, ensuring that a diagnosis is always produced.

5.2.1 Concordant and Discordant Elements

In STASEL, concordant elements refer to grammatical entities that

1. show no excess structure; that is, elements that are identified as *simple* in the structural analysis of the SSA module;

2. relate to the rest of the sentence without ambiguity of attachment. Examples of entities that attach without ambiguity are the functional entities of a main clause (subject, complements, verb modifiers) and dependent clauses with a subject, such as conditional clauses and appositions.

Discordant elements fall under two classes: each discordant element is defined as being either (simply) discordant (`discord-1`) or highly discordant (`discord-2`). This distinction is made to increase the subtlety of the resolution and positive interruption rules: A `discord-1` element can be part of a resolution or act as an interrupting element, but the mere presence of a `discord-2` element has a "contagious" discordant effect, similar to the detection of a *complex* entity in the structural analysis, in that the entire sentence becomes immediately unclear.

To determine which type of discord is associated with a given element, the GSA module first looks at the complexity analysis provided in the structural analysis. If a complex entity is present, the element involved automatically becomes a `discord-2` type and the entire sentence is diagnosed as being unclear. If no complex entity is detected, and if the element being examined is a *modifying* element (that is, not part of the main clause), the GSA module also consults a table that associates each modifying element with its concordance attribute (either concordant or `discord-1`) according to its potential ambiguity of attachment. `Discord-1` modifying elements can either be

- a clause without a subject, such as a prepositional phrase, a participial phrase or a dependent infinitive clause (as shown below):

  (5.13) To judge from the canvases on the walls of the gallery, those who are responsible are myopic.

- an element with little structure and no subject, such as a simple adjectival phrase:

  (5.10) John came to the palace, very happy.

  Other elements that are discordant because of extreme conciseness in structure are simple adverbs, infinitive verbs with no post-modification, and simple gerunds.

- an unusual construction, such as an inverted conditional clause:

  (5.11) Had John only known about the examination, he would have come.

5.2.2 Complex Entities

A complex entity stands for a syntactic construct that either displays excess structure, or whose organization leads to ambiguity of reading. The most common complex structures detected by STASEL and identified with the label `complex >>` [6] in the structural analysis of the SSA module are listed below:

- prepositional phrase followed by a dependent clause:

  (5.14) Surely, I will definitely never personally show you the solution of the problem that was given to the class.

  This structure is complex because it is not clear whether the dependent clause attaches to the noun modified by the prepositional phrase (*solution*) or to the object of the prepositional phrase (*problem*).

- prepositional phrase followed by a conjunction of noun phrases:

  (4.2) John bought books about computers and music.

  As seen earlier in Section 4.2, such constructions are structurally ambiguous because it is not possible to determine from the structure alone which of *computers and music* or *computers* is the object of the prepositional phrase.

- conjunction of noun phrases followed by a dependent clause:

  (5.1) The museum acquired paintings and canvases that were recycled.

  Such constructions are ambiguous because it is not clear whether the dependent clause refers to the compound noun phrase (*paintings and canvases*) or simply to the closest noun phrase (*canvases*).

- modifying element with ambiguous referent:

  (5.17) Politicians of this world, being rather nice, capture our interest.

  In such cases, the modifying element *being rather nice* has no clear referent: it could be *this world* that is nice, or *the politicians of this world.*

- three or more levels of nested prepositional phrases:

  (5.5) The brother of the cousin of my friend from class is also an engineer.

  An excessive number of nested prepositional phrases hinders the flow of a sentence and runs the risk of confusing the reader.

- three or more levels of embedded clauses:

  (5.6) It is unthinkable to move the whole of the works which are the only paintings in the palace where they were hung in the year the artist died.

[6] This label applies to the edited version of the structural analysis that is printed in the final output message of STASEL. In the internal representation passed to the GSA module, complex entities are labelled either with `complexNP` or `complex=`.

High levels of embedding tend to yield long and heavy sentences and make it difficult to follow the main idea of the sentence.

- excessive modification:

  (5.4) The computer power energy need of this lab is tremendous.

  Sentences can grow both long and confusing if too many modifiers are used.

- unbalanced organization in a compound structure:

  (5.21) Learning to speak a second language in a place where the language spoken is different can be difficult and writing it is almost impossible.

  In an unbalanced compound structure, the clauses on each side of the conjunction do not contribute the same weight in the overall arrangement, and thus, are often difficult to read and may fail to carry ideas clearly.

- excessive conjunction of noun phrases in the subject position:

  (5.7) Skating, dancing, and swimming are my favorite sports.

  High levels of coordination in the subject position add unnecessary weight to the sentence and may confuse the reader as to which elements are dependent rather than parallel.

Two examples will now be considered to illustrate how the GSA module performs its clarity diagnosis.

5.2.3 Monoschematic Clarity Rule

The clarity rule for monoschematic sentences reads as follows: a sentence consisting of an independent clause that, in turn, consists of a subject, complements, and verb modifiers, is clear with feature *monoschematic* if all of the following are true:

- the subject is a simple noun phrase;
- the subject is not a three-way conjunction of noun phrases;
- complements, if present, are simple in structure;
- verb modifiers, if present, are simple in structure;
- there is no excessive noun modification;
- there is no excessive qualification;
- there is no excess number of adverbs;
- there is no excess number of passive verbs.

In pseudo-PROLOG form, the rule reads as follows:

```
monoschematic( clause(Subject,Complements,Modifiers) ) :-
        simpleNP( Subject),
        isnot ( conjNP3(Subject) ),
        simpleC( Complements ),
        simpleM( Modifiers ),
        isnot( overMod(clause) ),
        isnot( overQ(clause) ),
        isnot( overADV(clause) ),
        isnot( overPASS(clause) ).
```

Consider the following sentence as input:

(5.1) The museum acquired paintings and canvases that were recycled.

Its structural analysis (internal version, Figure 5.1) shows that there is a complex element in the complement position, corresponding to an ambiguous attachment of the two-way conjunction of noun phrases (np2). Hence, the monoschematic clarity rule will fail at line 3 and the set of backup monoschematic rules will be invoked to diagnose the cause of the failure. The resulting diagnosis is printed in the final output message (Example 5.1) along with the structural analysis.[7]

```
clause(  subj(simpleNP(none)),
         comp(complexNP(np2)),
         noMod)
```

Figure 5.1 Structural analysis of Example 5.1

```
----------------------------------------------------------------------

ANALYSIS OF:  The museum acquired paintings and canvases that were
              recycled.
----------------------------------------------------------------------

 *** sentence is unclear ***

  >> complex structure in the predicate

----------------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: simple noun phrase (np)
        comp: complex >> conjunction of 2 noun phrases
----------------------------------------------------------------------
```

Example 5.1

[7] To save space and help focus on the clarity diagnosis, every output example presented in this chapter has been stripped of the syntactic style analysis (SSA) segment.

5.2.4 Centroschematic and Resolution Clarity Rules

The centroschematic and resolution rules read as follows:

- `centroschematic`: a sentence that consists of an initial modifying element (`IME`) followed by a main clause (`MainClause`) is clear with feature *centroschematic* if
  - the `IME` is concordant, and
  - the main clause is concordant (read clear) with feature *monoschematic.*
- `resolution`: a sentence consisting of an `IME` and a `MainClause` is clear with feature *resolution* if
  - the `IME` is discordant, and
  - the main clause is highly concordant (read extremely clear) with feature *restricted monoschematic.*

The following presents both rules in pseudo-PROLOG form:

```
centroschematic( IME + MainClause) :-
     concordant(IME),
     monoschematic(MainClause).


resolution( IME + MainClause) :-
     discord-1(IME),
     restricted_monoschematic(MainClause).
```

Consider Example 5.2 as an example. The sentence of Example 5.2 is a centroschematic type because it consists of an dependent phrase (called initial modifying element) followed by a main clause.

(5.2) Despite the weather and the road conditions, I returned safely.

Figure 5.2 shows the structural analysis of Example 5.2.

```
imc(modC=
        prepositional(pp2),

    clause(
         subj(simpleNP(none)),
         noComp,
         mod(adverb)))
```

Figure 5.2 Structural analysis of Example 5.2

A centroschematic sentence type is judged clear if all its elements are concordant (centroschematic rule) or if its initial discordant element is resolved by a highly concordant main clause (resolution rule). In other words, a sentence that is centroschematic in structure will always try both the centroschematic and resolution rules before turning to backup rules.

To judge the clarity of sentence 5.2, the GSA module first enters the centroschematic rule. While evaluating the concord of the initial modifying element (the prepositional phrase with a two-way compound noun phrase as object), it realizes that the element is not concordant and leaves the rule immediately. The resolution rule is then applied and the initial modifying element is found to be a discord-1 type, but the main clause is sufficiently clear (highly concordant) to resolve the initial discord.

```
----------------------------------------------------------------

ANALYSIS OF:  Despite the weather and the road conditions, I
              returned safely.
----------------------------------------------------------------

 <=> SENTENCE IS CLEAR: resolution

----------------------------------------------------------------

 <=> STRUCTURE: initial modifying component + clause

    initial modC -->
        pp that contains a 2-way conjunct np

    clause -->
        subj: simple noun phrase (np)
        mod: adverb
----------------------------------------------------------------
```

Example 5.2

5.3 Examples of Clarity Diagnosis in STASEL

This section presents examples of clarity diagnoses performed by STASEL and shows how the system provides instructional feedback to the student. If a sentence is judged to be structurally clear, STASEL informs the student of the diagnosis and states the name of the stylistic rule that was used in the analysis. If the sentence is somewhat clear, but contains a stylistic problem that, if corrected would contribute to a clearer structure, STASEL responds with a `NEUTRAL ANALYSIS` that explicitly states the stylistic problem encountered and provides remedial feedback. Otherwise, a message `*** sentence is unclear ***` is printed, followed by a list of reasons why the sentence lacks clarity. Since the structural analysis of the sentence is displayed below the clarity diagnosis, the student can correlate the reasons given by the system for the lack of clarity with precise sentence elements.

5.3.1 Monoschematic Feature

- Example 5.3 is structurally clear because none of the elements of its structural analysis are complex (which is synonymous with discordant);
- Example 5.4 illustrates the situation where the sentence is structurally clear, but would be clearer if it were not for the excessive noun modification.
- Example 5.5 demonstrates the effect of excess structure (too many levels of nested prepositional phrases) in the subject position;

- Example 5.6 demonstrates the effect of excess structure (too many levels of embedded clauses) in the predicate;
- Example 5.7 is a particular case of excess structure in the subject position (three-way compound noun phrase) that violates line 2 of the monoschematic clarity rule.

```
------------------------------------------------------------------

ANALYSIS OF:  I told him that she was wrong.

------------------------------------------------------------------

 <=> SENTENCE IS CLEAR: monoschematic

------------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: simple noun phrase (np)
        comp: simple noun phrase (np)
        comp: -- relative clause --
              clause -->
                  subj: simple noun phrase (np)
                  comp: adjectival phrase
------------------------------------------------------------------
```

Example 5.3

```
------------------------------------------------------------------

ANALYSIS OF:  The computer power energy need of this lab is
              tremendous.

------------------------------------------------------------------

 <=> SENTENCE IS CLEAR: monoschematic

       But it would be clearer without the
       excessive number of noun modifiers

------------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: np with excessive modification or qualification
        comp: adjectival phrase

------------------------------------------------------------------
```

Example 5.4

```
-------------------------------------------------------------------

ANALYSIS OF:  The brother of the cousin of my friend from class
              is also an engineer.

-------------------------------------------------------------------

 *** sentence is unclear ***

  >> complex structure in the subject

-------------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: complex >> np with excessive nesting of pp
        comp: simple noun phrase (np)

-------------------------------------------------------------------
```

Example 5.5

```
-------------------------------------------------------------------

ANALYSIS OF:  It is unthinkable to move the whole of the works
              which are the only paintings in the palace where
              they were hung in the year the artist died.

-------------------------------------------------------------------

 *** sentence is unclear ***

  >> complex structure in the predicate

-------------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: simple noun phrase (np)
        comp: adjectival phrase
        comp: complex >> infinitive with complex complement
-------------------------------------------------------------------
```

Example 5.6

```
-----------------------------------------------------------------

ANALYSIS OF:  Skating, dancing, and swimming are my favorite
              sports.

-----------------------------------------------------------------

 *** sentence is unclear ***

  >> excess structure in subject: conjunction of 3 np

-----------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: complex >> conjunction of 3 noun phrases
        comp: simple noun phrase (np)

-----------------------------------------------------------------
```

Example 5.7

5.3.2 Centroschematic Feature

- Example 5.8 is a clear centroschematic sentence with a concordant initial modifying element (a subordinate clause with a subject) and a concordant main clause. It is interesting to compare this example with the next output (Ex. 5.9) in which the same sentence is slightly modified, and with the first example of the resolution section (Ex. 5.12), where the subordinate clause is replaced with an infinitive clause.

- Example 5.9 is a modified version of Example 5.8 in which a third level of prepositional phrase is added to the initial modifying element. This modification translates into an excess structure (discord-2) and causes the centroschematic clarity rule to fail.

- Example 5.10 shows a centroschematic sentence with a *final* modifying element (adjectival phrase) that is discordant because it does not clearly attach to a specific part of the preceding main clause.

- Example 5.11 illustrates a situation in which a neutral analysis is diagnosed by STASEL. Neutral analyses are used to handle special cases where the syntactic structure evaluated by the system neither fulfils the requirements of a clarity rule nor displays features that lead to an "unclear" diagnosis. This example shows how the system treats inverted conditional structures.

5.3.3 Resolution

- Example 5.12 shows the result of a stylistic resolution. The initial modifying element is discord-1 because it lacks a subject, but it is resolved clearly by a highly concordant main clause. If this example is compared with Ex. 5.8, it is readily seen that the presence of discord in the former shifts the clarity analysis from a centroschematic "all-concordant" diagnosis to a somewhat less clear resolution.

```
-------------------------------------------------------------------

ANALYSIS OF:  If we can judge from the canvases on the walls,
              those who are responsible are myopic.

-------------------------------------------------------------------

 <=> SENTENCE IS CLEAR: centroschematic

-------------------------------------------------------------------

 <=> STRUCTURE: initial modifying component + clause

    initial modC -->
        -- subordinate clause --
              clause -->
                  subj: simple noun phrase (np)
                  mod: nested prepositional phrases

    clause -->
        subj: np with relative clause
        comp: adjectival phrase
-------------------------------------------------------------------
```

Example 5.8

```
-------------------------------------------------------------------

ANALYSIS OF:  If we can judge from the canvases on the walls of
              the gallery, those who are responsible are myopic.

-------------------------------------------------------------------

 *** sentence is unclear ***

  >> discordant element in initial position

-------------------------------------------------------------------

 <=> STRUCTURE: initial modifying component + clause

    initial modC -->
        -- subordinate clause --
              clause -->
                  subj: simple noun phrase (np)
                  mod: complex >> excessive nesting of pp

    clause -->
        subj: np with relative clause
        comp: adjectival phrase

-------------------------------------------------------------------
```

Example 5.9

```
-----------------------------------------------------------------

ANALYSIS OF:  John came to the palace, very happy.

-----------------------------------------------------------------

 *** sentence is unclear ***

  >> discordant element in final position
-----------------------------------------------------------------

 <=> STRUCTURE: clause + final modifying component

    clause -->
        subj: simple noun phrase (np)
        mod: simple prepositional phrase (pp)

    final modC -->
        adjectival phrase

-----------------------------------------------------------------
```

Example 5.10

```
-----------------------------------------------------------------

ANALYSIS OF:  Had John only known about the examination, he would
              have come.

-----------------------------------------------------------------

  NEUTRAL ANALYSIS
  >> complex inverted conditional structure

       The sentence is neutrally clear because the
       use of inverted structures is emphatic and
       highly formal. Such constructions should not
       be used often.

-----------------------------------------------------------------

 <=> STRUCTURE: initial modifying component + clause

    initial modC -->
        -- inverted conditional clause --
              clause -->
                  subj: simple noun phrase (np)
                  mod: simple prepositional phrase (pp)

    clause -->
        subj: simple noun phrase (np)

-----------------------------------------------------------------
```

Example 5.11

- Example 5.13 is a modified version of Example 5.12 in which a third prepositional phrase has been added to the initial modifying element. This modification translates to an excess structure (discord-2) in the infinitive clause, and causes the *centroschematic* rule to fail automatically, even before the resolution rule is entered.
- Example 5.14 is a very poorly written sentence that exhibits two structural problems, causing the failure of the resolution rule: an excess number of adverbs and an ambiguous relative clause attachment.

```
-----------------------------------------------------------------

ANALYSIS OF:  To judge from the canvases on the walls, those who
              are responsible are myopic.
-----------------------------------------------------------------

 <=> SENTENCE IS CLEAR: resolution

-----------------------------------------------------------------

 <=> STRUCTURE: initial modifying component + clause

    initial modC -->
        infinitive clause

    clause -->
        subj: np with relative clause
        comp: adjectival phrase
-----------------------------------------------------------------
```

Example 5.12

```
-----------------------------------------------------------------

ANALYSIS OF:  To judge from the canvases on the walls of the
              gallery, those who are responsible are myopic.
-----------------------------------------------------------------

 *** sentence is unclear ***

  >> discordant element in initial position

-----------------------------------------------------------------

 <=> STRUCTURE: initial modifying component + clause

    initial modC -->
        complex >> infinitive with complex verb modifier

    clause -->
        subj: np with relative clause
        comp: adjectival phrases
-----------------------------------------------------------------
```

Example 5.13

```
-----------------------------------------------------------------
ANALYSIS OF:  Surely, i will definitely never personally show you
              the solution of the problem that was given to the
              class.
-----------------------------------------------------------------
 *** sentence is unclear ***

  >> failed resolution
  >> excessive number of adverbs
  >> complex structure in the predicate
-----------------------------------------------------------------
 <=> STRUCTURE: initial modifying component + clause

    initial modC -->
        adverb

    clause -->
        subj: simple noun phrase (np)
        comp: simple noun phrase (np)
        comp: complex >> embedded structure in pp
-----------------------------------------------------------------
```

Example 5.14

5.3.4 Positive Interruption

- Example 5.15 demonstrates a "strong" positive interruption in which both the interrupting element and the main clause are concordant.

- Example 5.16 demonstrates a "weak", yet still successful positive interruption. The interrupted sentence of this example is not as clear as Ex. 5.15 because the interrupting element is discordant (`discord-1`). The highly concordant nature of the surrounding main clause, and particularly the simple noun phrase subject, permit the positive resolution of the interruption.

- Example 5.17 shows the same sentence as Ex. 5.16 but with the subject changed to a noun phrase modified by a prepositional phrase. With this increase in structure, the main clause is no longer highly concordant and the interrupting element no longer attaches non-ambiguously to the subject. Hence, the positive interruption rule fails.

```
------------------------------------------------------------------

ANALYSIS OF:  The famous collector, when he was unable to acquire
              certain canvases, bought copies.

------------------------------------------------------------------

 <=> SENTENCE IS CLEAR: positive interruption

------------------------------------------------------------------

 <=> Structure: simple clause with interruption

    interruption -->
        -- adverbial clause --
              clause -->
                  subj: simple noun phrase (np)
                  comp: adjectival phrase
                  comp: infinitive clause

    clause -->
        subj: simple noun phrase (np)
        comp: simple noun phrase (np)

------------------------------------------------------------------
```

Example 5.15

```
------------------------------------------------------------------

ANALYSIS OF:  Politicians, being rather nice, capture our interest.

------------------------------------------------------------------

 <=> SENTENCE IS CLEAR: positive interruption

------------------------------------------------------------------

 <=> Structure: simple clause with interruption

    interruption -->
        gerundival clause

    clause -->
        subj: simple noun phrase (np)
        comp: simple noun phrase (np)

------------------------------------------------------------------
```

Example 5.16

```
-------------------------------------------------------------------

ANALYSIS OF:  Politicians of this world, being rather nice,
              capture our interest.

-------------------------------------------------------------------

 *** sentence is unclear ***

  >> failed positive interruption
  >> discordant element in interrupting clause

-------------------------------------------------------------------

 <=> Structure: simple clause with interruption

    interruption -->
        gerundival clause

    clause -->
        subj: np with simple prepositional phrase (pp)
        comp: simple noun phrase (np)

-------------------------------------------------------------------
```

Example 5.17

5.3.5 Compound Structure

- Example 5.18 exhibits a compound structure of independent monoschematic clauses that is clear, parallel, and balanced. The structural analysis of the example clearly shows that both clauses display the same grammatical form.

- Example 5.19 shows a compound structure of two balanced verb phrases. This example also demonstrates the flexibility of STASEL's analysis, as it can be seen in the structural analysis that the complement positions of both clauses are not filled by exactly identical structures. Indeed, the system will only be concerned with unbalanced structure if pairs of corresponding elements are clearly not parallel.

- Example 5.20 displays an unbalanced compound structure of sentences. Again, the structural analysis clearly shows which of the functional entities of the sentence violates the parallelism principle (the complements). As no other structural problems have been detected, the resulting analysis is neutral.

- Example 5.21 shows an unclear compound structure of sentences that is both unbalanced and discordant because of the presence of a complex entity in the subject position of the first clause.

```
-----------------------------------------------------------------

ANALYSIS OF:  The style was formed and the principles were acquired.

-----------------------------------------------------------------

 <=> SENTENCE IS CLEAR: compound structure

-----------------------------------------------------------------

 <=> STRUCTURE: compound of sentences

    first sentence:
        clause -->
            subj: simple noun phrase (np)


    second sentence:
        clause -->
            subj: simple noun phrase (np)

-----------------------------------------------------------------
```

Example 5.18

```
-----------------------------------------------------------------

ANALYSIS OF:  My brother enjoys swimming but likes putting his
              head in the water.

-----------------------------------------------------------------

 <=> SENTENCE IS CLEAR: compound structure

-----------------------------------------------------------------

 <=> STRUCTURE: compound of verb phrases

    compoundVP -->
        subj: simple noun phrase (np)

        first vp:
            comp: gerund
        second vp:
            comp: gerundival clause

-----------------------------------------------------------------
```

Example 5.19

```
-----------------------------------------------------------------

ANALYSIS OF:  My brother enjoys swimming and my brother is also
              my best friend.

-----------------------------------------------------------------

  NEUTRAL ANALYSIS
  >> unbalanced compound structure

       The sentence is neutrally clear because the
       elements of the first clause are not equivalent
       in form to the elements of the second clause

-----------------------------------------------------------------

 <=> STRUCTURE: compound of sentences

    first sentence:
        clause -->
            subj: simple noun phrase (np)
            comp: gerund

    second sentence:
        clause -->
            subj: simple noun phrase (np)
            comp: simple noun phrase (np)
-----------------------------------------------------------------
```

Example 5.20

```
-----------------------------------------------------------------

ANALYSIS OF:  Learning to speak a second language in a place
              where the language spoken is different can be
              difficult and writing it is almost impossible.

-----------------------------------------------------------------

 *** sentence is unclear ***

  >> complex structure in the subject of the first sentence
  >> unbalanced compound structure

-----------------------------------------------------------------

 <=> STRUCTURE: compound of sentences

    first sentence:
        clause -->
            subj: complex >> gerundival with complex complement
            comp: adjectival phrase

    second sentence:
        clause -->
            subj: gerundival clause
            comp: adjectival phrase
-----------------------------------------------------------------
```

Example 5.21

Chapter 6

Concluding Remarks

This chapter reviews the contributions and limitations of the research described in this thesis and lays the foundation for future research.

The preceding chapters presented an overview of the field of intelligent computer-assisted language instruction (ICALI), and introduced a prototype system, STASEL, that analyzes input sentences interactively for the purpose of language instruction. The system is based on natural language parsing techniques and is specifically developed for instruction in linguistic style. STASEL's stylistic processing capabilities reside in two separate modules: a syntactic style analyzer (SSA) module that detects stylistic problems in diction, usage, and sentence structure, and a goal-directed style analyzer (GSA) module that determines the structural clarity of the input. The system summarizes its findings in a structured output message that gives an explanation of the stylistic errors encountered and provides remedial feedback. The content of the output message can easily be tailored to the requirements of particular teaching methods by the proper pedagogical authorities. A message is displayed to the student after each interaction.

6.1 Contributions

STASEL is of interest not only because it is based on a comprehensive parser/stylistic analyzers combination that recognizes an important range of syntactic structures and stylistic features of sentences, but also because it performs *practical* goal-directed analysis of the input according to formal principles of stylistic clarity. The subtlety and exhaustiveness of the clarity analysis performed by STASEL (demonstrated in Chapter 5) show that sophisticated goal-directed processing is not the abstract and informal concept that many believe it to be, but a worthwhile application that can be effectively computerized.

This research demonstrates that both syntactic and goal-directed principles of style can be practically and successfully implemented for the purpose of language instruction. This is achieved in STASEL by

- using a parsing approach that recognizes stylistic as well as syntactic features in the input sentence;

- generating explicit representations of the sentence's elements in the form of descriptive analyses that are specifically suited for stylistic processing;

- formally encoding the rules and conventions of English style in the analyzer modules;
- providing a communicative environment where the student can "learn by doing".

The descriptive analyses generated at various stages in the system, notably the stylistic parse tree (Section 3.1.2) and the high-level structural analysis of the SSA module (Section 4.2) are particularly significant. They offer a novel and flexible means of treating various kinds of stylistic considerations and facilitate the refinement and extension of existing features. This flexibility is readily apparent in the case of the structural analysis, which serves both as input to the GSA module and as instructional output in the message displayed to the student.

The descriptive analyses not only provide the preliminary processing that is fundamental to the detection of a wide (and expandable) set of stylistic problems, but also carry the basic information contained in the input sentence in a form that is tailored to the needs of subsequent stages. This technique contributes to a more controlled flow of information through the system, and allows for increased modularity and easy expansion in the design.

Providing a high degree of expandability was a central concern in the design of STASEL and was dictated by the long term objective of this research: that STASEL will form the basis for the development of intelligent tutoring systems for teaching writing (as illustrated in Figures 1.2 and 1.3).

6.2 Limitations and Future Directions

Although STASEL covers a wide range of stylistic problems, the system was not intended to embody all the characteristics of a fully functional stylistic tutor, but rather to provide a structured core around which a pedagogically sound, highly communicative intelligent system can be built. In its present form, the prototype has limitations in the following domains:

- *Goal-directed coverage*
  A system that performs goal-directed stylistic analysis must be able to analyze more than one goal. Hence, additional goals need to be incorporated in the GSA module. Examples of possible future goals are *emphasis*, where the student is taught how to put stress on given sentence elements and how to use strategic emphatic positions within a sentence, and *formality*, where the student is introduced to the syntactic structures and sentence constructions that are associated with a highly formal and conservative style.
- *Semantic processing*
  As appropriate techniques are developed, semantic information will need to be added in the lexicon, in the parser, and in the stylistic analyzers to resolve some of the deficiencies of the current implementation. Semantic processing capabilities will help speed up the parsing process by detecting ambiguities that arise from context and increase the coverage of stylistic problems, such as misplaced and dangling modifiers (Section 4.3.3), that require semantic information for their resolution. Semantics is also essential to enhance the clarity analysis of the GSA module, since a sentence that does not make sense, no matter how well its elements have been put together, will always remain unclear.

- *Parser functionality*
  Section 3.3 showed that there are still several complicated structures that STASEL's parser does not recognize. Additional phrase structure rules are needed to parse complex structures, and allow greater flexibility in input sentences.

- *Parser speed*
  As shown in Section 3.3, a backtracking parser may cause a significant and sometimes unacceptable increase in response time when analyzing unrecognizable, complex, or ambiguous structures. STASEL's parser is not sufficiently robust or efficient to be used in a practical context.

- *Stylistic processing*
  Some of the messages generated by the SSA module could benefit from some refinements. These include recurring and pointlessly repetitive messages such as `* 2 passives detected` and `WARNING: vague word "nice"`, and the incorrect flagging of word choice caused by STASEL's inability to distinguish between multiple definitions of the same word. For example, if the word *class* is identified as an abstract word in the lexicon, then all occurrences of the word will be flagged as being abstract, even in the cases where it is used to mean *a class of students.*

- *Lexicon content*
  STASEL's lexicon is too small (about 800 items) for practical use. The lexicon should eventually include a full dictionary that will not only be used to recognize lexical items during parsing, but also be made available for online consultation.

- *User interface*
  To be practical in a classroom, STASEL needs an improved user interface with a comprehensive help facility, a more detailed and meaningful output display, and perhaps some customizing devices to allow the student to choose the types of stylistic messages, or even goals, he would like to investigate.

Despite these limitations, the design of STASEL presents a highly modular approach and introduces a variety of techniques in syntactic and goal-directed stylistic processing that can serve as a basis for expansion. The next step is to build upon the principles and techniques developed in STASEL and implement the components of the ITS architecture for writing (described in Section 1.4) that are not covered by the present prototype. This would involve

- improving the natural language interface by adding an intelligent grammar corrector that can correct spelling and grammatical errors in the input sentence (see works by Richardson and Braden-Harder [RBH88] and Catt [Cat88]);
- including a rhetoric and pragmatics module that assesses the input sentence according to the writer's intention or purpose (see work by Hovy [Hov87]);
- building a student model that contains information on the student's presumed knowledge, past performance, and level of ability, and that will interact with the other components of the system to provide personalized instruction (see works by Self [Sel87], Cohen and Jones [CJ86], Kass and Finin [KF88]);
- expanding the pedagogical component to include help facilities and online tutorial texts that provide formal instruction and that the student can consult at will.

- providing an explanation planner to tailor the content of explanations to the student's level of ability (see works by Paris [Par88] and Brecht [Bre91], [BJMG89]);
- devising tutorial strategies in collaboration with education professionals;
- integrating a natural language generator, so that responses are not "canned" (see work by Evans and Levin [EL89]).

This research was premised on the notion that teaching stylistic rudiments, far from being an elusive concept, is a task that can be performed with the help of instructional computer programs. Another premise of this research was that artificial intelligence techniques offer a practical and communicative means by which to interact with students. I believe STASEL has met both expectations.

Appendix A

Glossary of Terms

For those who may not be familiar with linguistic and stylistic terminology, this section provides definitions of frequently used terms divided in two categories: (1) grammar of the sentence and (2) stylistic vocabulary. The definitions marked by an asterisk ("*") are re-printed from the output of the help facility available on STASEL.

Some of the system-generated definitions are introduced by a characteristic key word that further categorize the word being defined. These key word are

- **USAGE –>**
  To introduce categories of words that are restricted to certain usage conventions, such as slang or archaic words.

- **Clarity Feature>**
  To define the stylistic shape and effect of a particular sentence construction. Such definitions are part of the stylistic vocabulary introduced in Chapter 5.

As in all attempts at categorizing and defining elements of linguistics, these definitions represent one way of looking at language, not necessarily shared by all linguists, but suitable for teaching purposes and consistent with the terminology used in this work. Writing and grammar texbooks by Hefferman [HL82], Frank [Fra86], Kane [Kan83], and Fowler [Fow61] were used as reference.

A.1 Grammar of the Sentence

adjectival phrase/clause

A phrase or clause that modifies a noun, describing complicated ideas that generally cannot be expressed in a single word or adjective. Adjectival phrases can be prepositional phrases,

> The book on the table belongs to John.

while adjectival clauses are usually introduced by a relative pronoun (as in a relative clause),

> The man who is waving at us is my brother.
> The place where we live is very cold.

or an adverb.

> The reason why he came escapes me.

adverbial clause

A clause that is used as an adverb within a sentence. It is introduced by a subordinating conjunction such as *because*, *since*, *when* or *although*. Adverbial clauses commonly modify another clause.

> Although he was tired, he decided to go on.

apposition

A word, phrase, or clause that is used to identify or describe another noun or term. Appositions are usually set off from the matrix clause by commas:

> John, my brother who is tall, is an engineer.

The skillful use of appositions is one sign of mature, sophisticated writing.

clause

A word group that consists of a subject (or noun phrase) and a predicate (also called verb phrase). There are two types of clauses: independent and dependent.

An independent clause can stand by itself as a simple sentence. The following example is an independent clause with subject *John* and predicate *went to the market.*

> John went to the market.

A dependent clause cannot stand alone as a complete sentence. It must be connected to or included in a main clause. The example below is a dependent clause introduced by the subordinating conjunction *if*, with subject *he* and predicate *had known about the examination.*

> If he had known about the examination, ...

complement

The complement is what follows the verb, "completing" the action or state described by the verb. A complement may take several forms, such as:

- *Direct object*: the person or thing directly affected by the action specified by the verb.

  I bought a book.

- *Indirect object*: The person or thing indirectly affected by the action performed on the direct object.

  I bought John a book.

- *Adjectival object*: A description of the state introduced by the verb.

  She looks pale.

- *Prepositional phrase object*:

  I give to the poor.

Complements usually follow the verb, though occasionally, for reasons of emphasis, they can be appear before the verb.

Hypocrisy, I despise.

complex sentence

A complex sentence consists of one independent clause and at least one dependent clause. The independent clause is usually called the main clause.

If he had known about the examination, he would have come early.
⊢ dependant. clause ⊣ ⊢ main clause ⊣

compound sentence

A sentence which consists of two or more independent clauses, commonly joined by a conjunction such as *and*, *but*, *or*, and *yet*.

John won the lottery but he gave it all to his mother.

conditional sentence

A sentence which first states a condition and provides the result of this condition in the main clause.

If it rains, the picnic will be cancelled.

copula verb

A verb such as *be*, *seem*, or *taste* that is used merely to identify or link the subject with the complement of a sentence. Copulas may serve to link nouns,

He is an engineer.

adjectival complements,

Sugar tastes sweet.

or adverbial complements.

John is in prison.

dependent clause see **clause**

expletives

Pronouns such as *it, this, that, there* which fill the subject position, replacing or anticipating the nominal subject.

It was necessary to stop the search.
There exist an infinite number of stars.

See **anticipatory construct** for the stylistic impact of expletives.

fragment

An incomplete sentence used as if it where a whole one.

She collapsed suddenly. And died.

gerund

The present participle form (*-ing* form) of a verb which acts as a noun or as a modifier within a sentence.

Skating is fun.

gerundival phrase

A phrase which consists of a gerund accompanied with other words acting as subject, complements or modifiers of the gerund. Gerundival phrases function as nouns within a sentence.

She worries about failing her exams.
His skating has improved over the years.

independent clause see **clause**

infinitive phrase

A phrase consisting of a single infinitive verb or an infinitive verb followed by complements and/or modifiers.

I like to skate.
To tell you the truth, I have had enough.

main clause

The independent clause in a complex sentence.

Concerned for the health of her baby, she went to the hospital.

main verb

(1) The verb which describes the action (or state) taken by the subject in a sentence.

The computer that she bought works very well.

(2) The verb of the independent clause in a complex sentence.

Concerned for the health of her baby, she went to the hospital.

modifier

A word or phrase that describes and clarifies the meaning of other terms.

My grey cat caught a little mouse.
He skates elegantly.
John went to the market.

noun phrase *

A pronoun, a proper name, or a phrase formed by a noun and its modifiers. Example noun phrases are:

We live in a castel.
John gave them a book.
In Great Britain, there are many tall men.
The technician wrote an efficient computer program.
Workers concerned with safety shut down the three machines.

participial phrase

A phrase which consists of a verb in participial form (either present or past participle) accompanied by its modifiers and complements. Participial phrases act as modifiers within a sentence.

Concerned for the health of her baby, she went to the hospital. (*past*)
They went dancing under the stars. (*present*)

phrase

A word group that does not contain a verb and functions as a unit, but lacks a subject, or a predicate, or both. Such groups act as nouns or as modifiers within a sentence.

There are seven types of phrases: adjectival, adverbial, gerundival, infinitive, noun, participial and prepositional

A verb phrase is a special type of phrase which contains a finite verb, but lacks a subject. "Verb phrase" is synonymous to the term "predicate".

predicate see **verb phrase**

prepositional phrase *

A phrase that consists of a preposition (such as *of, to, by, in, with*, etc) followed by its object:

I like coffee with milk.
The cat of my dreams is black.

relative clause

A subordinate clause that modifies a noun within a sentence. Also called adjectival clause, it is usually introduced by a relative pronoun (*that, who, which* etc.):

The man who came yesterday wants to buy the company.

subject

The subject of a sentence consists of a word or a word group that tells who or what performs or undergoes the action, or who or what experiences the condition named by the verb. A subject usually precedes the verb, though occasionally, for reasons of emphasis, it can be appear after the verb.

John and Mary work at the university.
With much precaution, the detonator was removed from the box.
If it rains, the picnic will be cancelled.
Come you with me.

subordinate clause

A type of dependent clause which is introduced by a linking word such as a relative pronoun (*who, that, which* etc.), or a subordinating conjunction (such as *before, because, while, when, although, as* or *since*).

Before we left, I locked all the doors.

subordinating conjunction see **subordinate clause**

verb phrase *

A verb or a verb group that describes the action performed, undergone, or experienced by the subject. Verb phrases are also called predicates. A verb phrase consists of a finite verb (with or without auxiliaries depending on the tense), which may or may not be followed by complements and modifiers:

John will never come to the party.
She had to give him the money.

A.2 Stylistic Vocabulary

abstract noun *

An abstract noun describes ideas and concepts rather than concrete objects. Examples: *structure, thought, definition.*

The extensive or excessive use of abstractions in composition leads to dullness and confusion. Abstractions are not inherently wrong, but they are often unnecessary even when dealing with ideas. The writer can restrict the use of abstract nouns by replacing them with participles whenever possible.
Examples:

abstract> I left with the thought that she was safe.
participle> I left thinking that she was safe.

anticipatory construct *

An anticipatory construct is created when a pronoun that has no meaning of its own is used to fill the subject position. Such constructions are usually introduced by one of these pronouns *it, this, these, those, there*

followed by a verb such as *to be* or *to seem*, used to link the pronoun to the main idea of the sentence:

there is a man in the office.

Anticipatory constructions use more words than comparable direct statements. Sometimes, the extra words serve a valid purpose, for emphasis or idiom, but when they do not, the writer should recast the sentence to make better use of its main elements.

archaic term *

USAGE ->
A word or expression that is no longer in common use but may be found in literary works or used to impart a historical colour to contemporary writing.

centroschematic *

Clarity Feature> A sentence that consists of a single dominant component (concordant main clause) accompanied by a concordant modifying element.

cliché *

USAGE ->
A word or expression that has lost much of its force through overexposure. Cliches are dull and unoriginal.

colloquial term *

USAGE ->
A word or expression that is appropriate to a conversational level of usage, but sounds out of place in formal writing.

Colloquial, slang, and taboo expressions are not grammatical errors but they might interfere with the writer's ability to communicate with the reader. Such expressions should be avoided in formal writing, not merely because they are inelegant or undignified, but because they are generally short-lived and often vague.

compound structure *

Clarity Feature> A sentence that consists of the coordination of two balanced and parallel elements (elements that display equivalent grammatical forms).

concordant *

Stylistically, an element is concordant if it displays clarity and stability on its own, without ambiguity of attachment or structure.

deadwood *

Deadwood is a word that fulfils no function. Deadwood words convey no additional information and contribute in no significant way to the meaning of the sentence. Deadwood violates the virtue of concision and interfers with communication.

diction

The study of diction is concerned with *choosing* words in a sentence so that they carry the meaning intended by the writer. In diction, the use of certain words is restricted by usage conventions.

discordant *

Stylistically, an element is discordant if it produces conflict, incongruity, or ambiguity because it lacks structural clarity and/or because of its position within the sentence.

faulty parallelism

A stylistic error in which the parallel construction principle has been violated.

FAULTY: I like swimming and to skate.
REVISED: I like swimming and skating.
or
I like to swim and to skate.

functional entity *

A functional entity is a word or word group that has a specific function within a sentence. Functional entities describe the higher-level constituents of a sentence and are expressed as one or more grammatical syntactic structures known as **grammatical entities**. Possible functional entities include the subject and if present, the complements, verb modifiers, interrupting elements, and modifying elements of the sentence.

grammatical entity *

A grammatical entity is a syntactic structure that describes how words are grouped together in a sentence. A grammatical entity can be a noun phrase, a prepositional phrase, an adverb, an infinitive clause, a relative clause, etc. Grammatical entities are either combined or used separately to form the functional entities of the sentence.

interrupting element see **modifying element**

interruption *

An interrupted sentence is characterized by the insertion of an interrupting element, usually separated by commas, after the subject of the sentence. Examples:

apposition> John, my brother who is tall, is an engineer.
infinitive phrase> This man, to tell the truth, is ridiculous.
adverbial clause> This painting, when it is properly hung,
looks quite different.

jargon *

USAGE ->
Misused technical term or expression. Technical language, such as the highly specialized vocabulary used in a particular field, should be restricted to expert writing.

modifying element *

A modifying element or component is a subordinate and dependent structure, either a clause or a phrase, that provides additional information to the main core of the sentence. Modifying elements can be placed at the beginning or at the end of a sentence. Modifying elements that are inserted in the middle of a sentence are called interrupting elements. Examples:

initial adverb>	Surely, she will come.
conditional clause>	If he were here, he would find a solution.
participial phrase>	Concerned for her aunt, she went home.
final adjectival>	I saw a man in the office, dead.
subordinate clause>	I enjoy music, because it is precise.

monoschematic *

Clarity Feature> A sentence that consists of an independent main clause with a single dominant shape and no modifying elements.

parallel construction *

Items in a series are parallel if they have the same grammatical form:

parallel>	I like to skate and to swim.
non-parallel>	I like to skate and swimming.

The specific use of parallel constructions is to order, group, or link similar ideas, or to list the items of a series. Parallel structure offers a clear and flexible way of ordering ideas in a sentence.

passives *

Style can be made more direct, vigorous and concise if the writer avoids unnecessary passive verbs. There are occasions when passives are legitimate, but if used in excess, they slow down the rhythm of the sentence. Passives are useful to:

* help keep the focus of the sentence on someone or something important that is acted upon;

* describe an action when the writer does not know or does not care to say who or what performs it;

* put the subject of the action at the end of the clause where the writer can easily attach a long modifier.

pompous term *

USAGE ->
A pompous (or pretentious) word is a fancy one used where a common term can be as efficient. Using simpler words results in writing clearer sentences.

positive interruption *

Clarity Feature> An interrupted sentence in which the interrupting element is concordant, thus supporting the overall balance of the sentence.

resolution *

Clarity Feature> A sentence that consists of an initial discordant modifying element followed by a highly concordant monoschematic clause that resolves the initial incongruity. Note that a centroschematic construct is clearer than a resolution because all its elements are concordant (whereas a resolution contains an initial discord element).

slang term *

USAGE ->
A word or expression that is extremely informal and has the connotation of particular social situations.

split infinitives *

An infinitive verb is said to have been split if a word or a group of words, usually adverbs, have been inserted between the infinitive marker *to* and the verb root.

Infinitive splits weaken the sentence structure and may lead to confusion. It is better to move the splitting group at the end of the infinitive phrase:

split>	I would like to <u>fully</u> understand the problem.
revised>	I would like to understand the problem <u>fully</u>.

A one-word modifier split is acceptable nowadays to most readers. Yet even now, unless there is no other suitable place in the sentence, it is best to respect the no-split convention.

taboo expression *

USAGE ->
Words that are not acceptable in polite use.

Colloquial, slang, and taboo expressions are not grammatical errors but they might interfere with the writer's ability to communicate with the reader. Such expressions should be avoided in formal writing, not merely because they are inelegant or undignified, but because they are generally short-lived and often vague.

usage

Usage designates the conventions that determine *how* language is used in certain situations. Usage conventions are less restrictive than grammar rules, but they should be respected, as most of them contribute to clarity and economy of expression in writing.

vague words *

Words that are weak and vacuous generally carry little meaning. Such words have proper English senses but they are frequently inserted to fill a grammatical gap or to make the sentence appear to say something more generally significant than it really does Examples are: ***way, kind, type, sort, nice***, etc.

He skated in an elegant <u>way</u>.
Her <u>type</u> of complexion is pale.

weak words see **vague words**

Appendix B

Example Analyses

```
-----------------------------------------------------------------

ANALYSIS OF:  Great Britain opposes it and Holland has raised
              slightly less strenuous objections.

-----------------------------------------------------------------

 <=> GOOD: stylistically correct

-----------------------------------------------------------------

 <=> SENTENCE IS CLEAR: compound structure

-----------------------------------------------------------------

 <=> STRUCTURE: compound of sentences

    first sentence:
        clause -->
            subj: simple noun phrase (np)
            comp: simple noun phrase (np)

    second sentence:
        clause -->
            subj: simple noun phrase (np)
            comp: simple noun phrase (np)

-----------------------------------------------------------------
```

Example B.1

```
------------------------------------------------------------------

ANALYSIS OF:  To tell everybody is the best thing.

------------------------------------------------------------------

 WARNING: vague word "thing"

------------------------------------------------------------------

 <=> SENTENCE IS CLEAR: monoschematic

------------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: infinitive clause
        comp: simple noun phrase (np)

------------------------------------------------------------------
```

Example B.2

```
------------------------------------------------------------------

ANALYSIS OF:  The artist provides a dreamy background done in
              yellow and bistre brushstrokes.

------------------------------------------------------------------

   *  1 passive detected

  USAGE--> pompous expression "bistre"

------------------------------------------------------------------

 <=> SENTENCE IS CLEAR: monoschematic

------------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: simple noun phrase (np)
        comp: np with participial clause

------------------------------------------------------------------
```

Example B.3

```
-------------------------------------------------------------------

ANALYSIS OF:  Her aunt having left the room, I declared my
              passionate love for Celia.
-------------------------------------------------------------------

 <=> GOOD: stylistically correct

-------------------------------------------------------------------

 <=> SENTENCE IS CLEAR: centroschematic
-------------------------------------------------------------------

 <=> STRUCTURE: initial modifying component + clause

    initial modC -->
        participial clause with subject
              subj: simple noun phrase (np)
              part: gerundival clause

    clause -->
        subj: simple noun phrase (np)
        comp: np with simple prepositional phrase (pp)
-------------------------------------------------------------------
```

Example B.4

```
-------------------------------------------------------------------

ANALYSIS OF:  He skated in a way that seemed elegant.

-------------------------------------------------------------------

 [W] wordy construct around "way"
     In good writing, adjectives and adverbs should link as
     directly as possible to what they really modify.
     A better link is to replace "way" and the
     relative clause following it with an appropriate adverb.

  USAGE--> unless there is genuine doubt or uncertainty, seem is
           likely to be deadwood. It is meaningful in
                 "they seem exhausted"
           but probably dead in
                "the students seem to enjoy the lecture
-------------------------------------------------------------------

 <=> SENTENCE IS CLEAR: monoschematic
-------------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: simple noun phrase (np)
        mod: pp with relative clause
-------------------------------------------------------------------
```

Example B.5

```
-----------------------------------------------------------------

ANALYSIS OF:  Concerned with the spiritual good health of the
              inhabitants, she transferred her powers to the
              young followers.

-----------------------------------------------------------------

  *  1 passive detected

 WARNING: vague word "good"

-----------------------------------------------------------------

 <=> SENTENCE IS CLEAR: resolution

-----------------------------------------------------------------

 <=> STRUCTURE: initial modifying component + clause

    initial modC -->
        participial clause

    clause -->
        subj: simple noun phrase (np)
        comp: np with simple prepositional phrase (pp)

-----------------------------------------------------------------
```

Example B.6

```
-----------------------------------------------------------------

ANALYSIS OF:  I like skating, dancing and swimming.

-----------------------------------------------------------------

 <=> GOOD: parallel construct achieved

-----------------------------------------------------------------

 <=> SENTENCE IS CLEAR: monoschematic

-----------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: simple noun phrase (np)
        comp: conjunction of 3 noun phrases

-----------------------------------------------------------------
```

Example B.7

```
-----------------------------------------------------------------

ANALYSIS OF:  Dancing has a super effect on me.

-----------------------------------------------------------------

 [W] wordy construct with "have" and "effect"
     It is better to use an active verb derived from "effect"

  USAGE--> colloquial expression "super"

 [W] restrict the use of vacuous words such as "effect"

-----------------------------------------------------------------

 <=> SENTENCE IS CLEAR: monoschematic

-----------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: gerund
        comp: np with simple prepositional phrase (pp)

-----------------------------------------------------------------
```

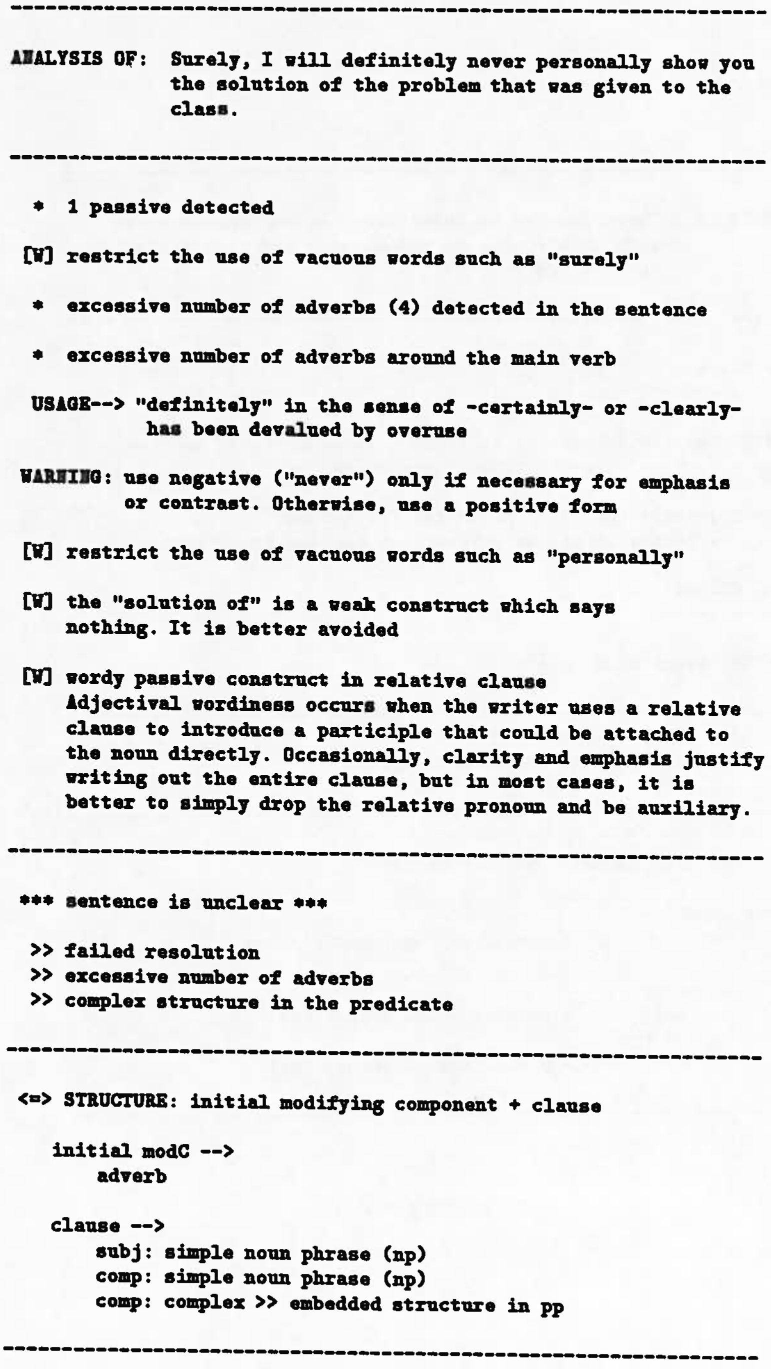
Example B.8

```
-----------------------------------------------------------------

ANALYSIS OF:  John bought music and books about computers.

-----------------------------------------------------------------

 <=> GOOD: acceptable parallel construction

-----------------------------------------------------------------

 <=> SENTENCE IS CLEAR: monoschematic

-----------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: simple noun phrase (np)
        comp: conjunction of 2 noun phrases

-----------------------------------------------------------------
```

Example B.9

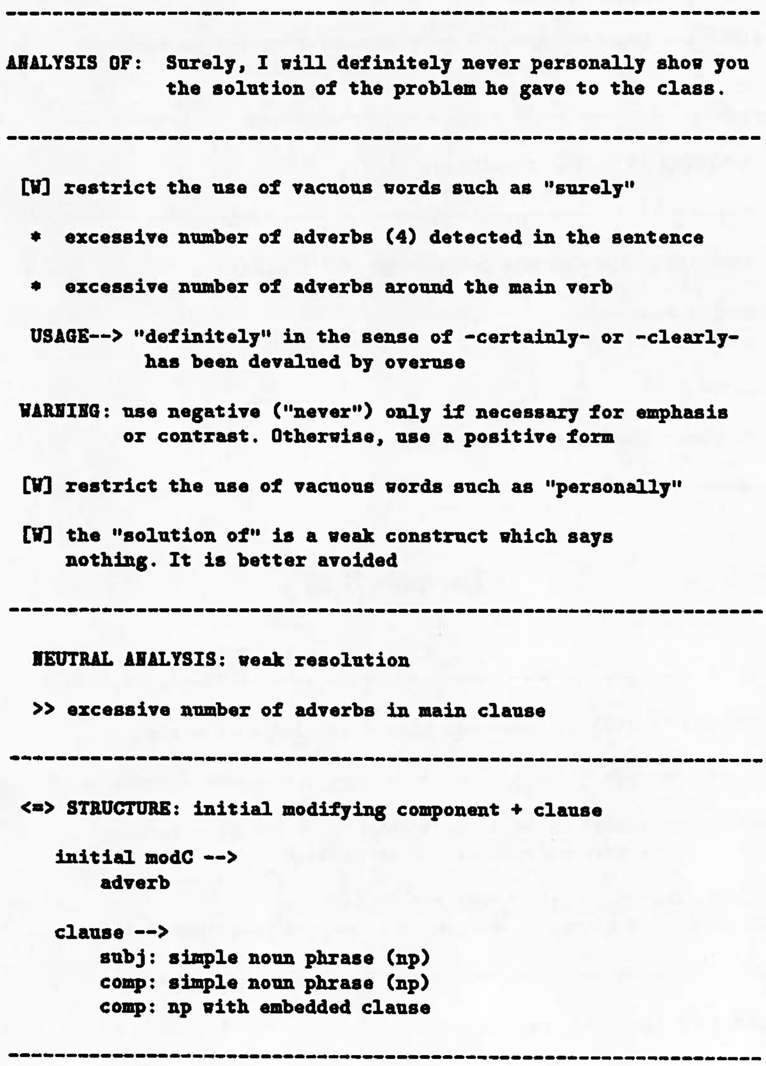
```
------------------------------------------------------------------

ANALYSIS OF:  Those who had an objection when our old director
              left should come to the meeting and vote against
              the new one.

------------------------------------------------------------------

First Clause:
--------------

 WARNING: as the subject of sentences, demonstratives such as
          "those" run the risk of vagueness

 [W] wordy construct with "have" and "objection"
     It is better to use an active verb derived from "objection"

Second Clause:
---------------

 WARNING: vague word "one"

------------------------------------------------------------------

 <=> SENTENCE IS CLEAR: compound structure

------------------------------------------------------------------

 <=> STRUCTURE: compound of verb phrases

    compoundVP -->
        subj: np with 2 levels of dependent clauses

        first vp:
            mod: simple prepositional phrase (pp)
        second vp:
            mod: simple prepositional phrase (pp)

------------------------------------------------------------------
```

Example B.10



```
------------------------------------------------------------------

ANALYSIS OF:  Surely, I will definitely never personally show you
              the solution of the problem that was given to the
              class.

------------------------------------------------------------------

  *  1 passive detected

 [W] restrict the use of vacuous words such as "surely"

  *  excessive number of adverbs (4) detected in the sentence

  *  excessive number of adverbs around the main verb

  USAGE--> "definitely" in the sense of -certainly- or -clearly-
           has been devalued by overuse

 WARNING: use negative ("never") only if necessary for emphasis
          or contrast. Otherwise, use a positive form

 [W] restrict the use of vacuous words such as "personally"

 [W] the "solution of" is a weak construct which says
     nothing. It is better avoided

 [W] wordy passive construct in relative clause
     Adjectival wordiness occurs when the writer uses a relative
     clause to introduce a participle that could be attached to
     the noun directly. Occasionally, clarity and emphasis justify
     writing out the entire clause, but in most cases, it is
     better to simply drop the relative pronoun and be auxiliary.

------------------------------------------------------------------

 *** sentence is unclear ***

  >> failed resolution
  >> excessive number of adverbs
  >> complex structure in the predicate

------------------------------------------------------------------

 <=> STRUCTURE: initial modifying component + clause

    initial modC -->
        adverb

    clause -->
        subj: simple noun phrase (np)
        comp: simple noun phrase (np)
        comp: complex >> embedded structure in pp

------------------------------------------------------------------
```

Example B.11



```
-----------------------------------------------------------------

ANALYSIS OF:  Surely, I will definitely never personally show you
              the solution of the problem he gave to the class.

-----------------------------------------------------------------

 [W] restrict the use of vacuous words such as "surely"

  *  excessive number of adverbs (4) detected in the sentence

  *  excessive number of adverbs around the main verb

  USAGE--> "definitely" in the sense of -certainly- or -clearly-
           has been devalued by overuse

 WARNING: use negative ("never") only if necessary for emphasis
         or contrast. Otherwise, use a positive form

 [W] restrict the use of vacuous words such as "personally"

 [W] the "solution of" is a weak construct which says
     nothing. It is better avoided

-----------------------------------------------------------------

  NEUTRAL ANALYSIS: weak resolution

  >> excessive number of adverbs in main clause

-----------------------------------------------------------------

 <=> STRUCTURE: initial modifying component + clause

    initial modC -->
        adverb

    clause -->
        subj: simple noun phrase (np)
        comp: simple noun phrase (np)
        comp: np with embedded clause

-----------------------------------------------------------------
```

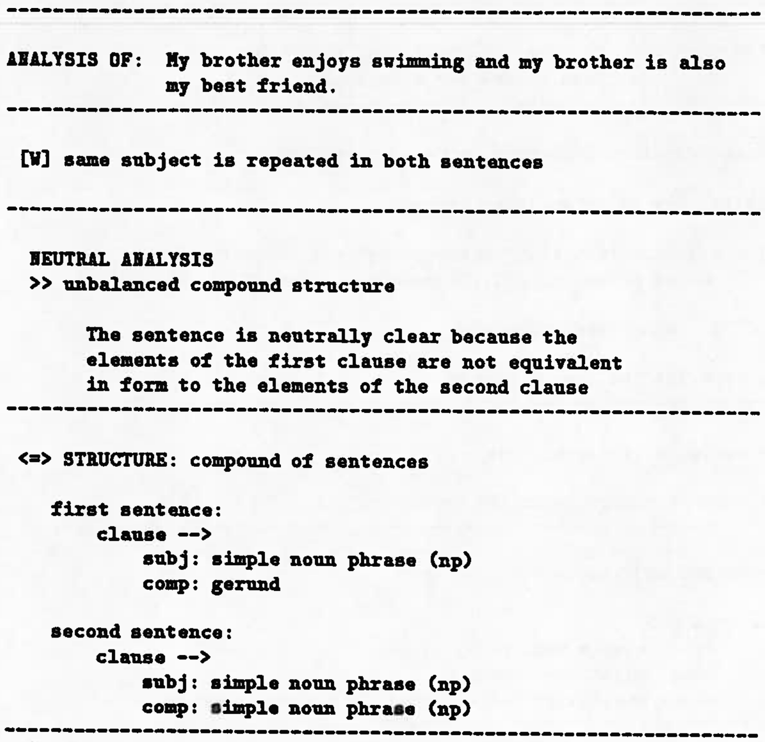
Example B.12

```
-----------------------------------------------------------------

ANALYSIS OF:  Despite the fact that he comes from Canada, he does
              not understand world politics.

-----------------------------------------------------------------

 [W] it is more concise to use "although" or "even though" instead
     of "despite the fact that"

 WARNING: use negative ("not") only if necessary for emphasis or
          contrast. Otherwise, use a positive form

-----------------------------------------------------------------

 <=> SENTENCE IS CLEAR: resolution

-----------------------------------------------------------------

 <=> STRUCTURE: initial modifying component + clause

    initial modC -->
        embedded clause

    clause -->
        subj: simple noun phrase (np)
        comp: simple noun phrase (np)

-----------------------------------------------------------------
```

Example B.13

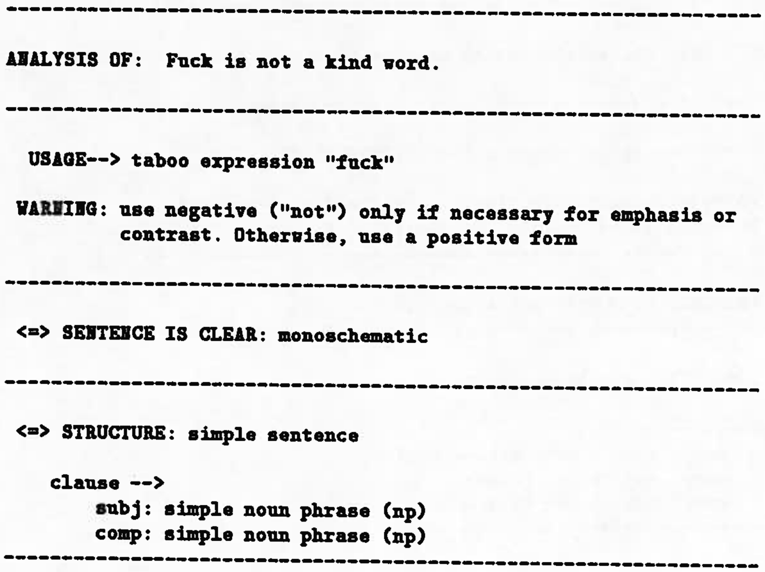
```
-----------------------------------------------------------------

ANALYSIS OF:  Actually, these men have a knowledge of music.

-----------------------------------------------------------------

  USAGE--> "actually" is an emphatic term that has been devalued
           by overuse - It is better avoided

 [W] wordy construct with "have" and "knowledge"
     It is better to use an active verb derived from "knowledge"

-----------------------------------------------------------------

 <=> SENTENCE IS CLEAR: resolution

-----------------------------------------------------------------

 <=> STRUCTURE: initial modifying component + clause

    initial modC -->
        adverb

    clause -->
        subj: simple noun phrase (np)
        comp: np with simple prepositional phrase (pp)
-----------------------------------------------------------------
```

Example B.14



```
-----------------------------------------------------------------

ANALYSIS OF:  My brother enjoys swimming and my brother is also
              my best friend.
-----------------------------------------------------------------

 [W] same subject is repeated in both sentences

-----------------------------------------------------------------

  NEUTRAL ANALYSIS
  >> unbalanced compound structure

       The sentence is neutrally clear because the
       elements of the first clause are not equivalent
       in form to the elements of the second clause
-----------------------------------------------------------------

 <=> STRUCTURE: compound of sentences

   first sentence:
       clause -->
           subj: simple noun phrase (np)
           comp: gerund

   second sentence:
       clause -->
           subj: simple noun phrase (np)
           comp: simple noun phrase (np)
-----------------------------------------------------------------
```

Example B.15



```
-----------------------------------------------------------------

ANALYSIS OF:  Fuck is not a kind word.

-----------------------------------------------------------------

  USAGE--> taboo expression "fuck"

 WARNING: use negative ("not") only if necessary for emphasis or
         contrast. Otherwise, use a positive form

-----------------------------------------------------------------

 <=> SENTENCE IS CLEAR: monoschematic

-----------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: simple noun phrase (np)
        comp: simple noun phrase (np)
-----------------------------------------------------------------
```

Example B.16

```
-----------------------------------------------------------------

ANALYSIS OF:  It is very interesting to show that some people are
              selfish in the way they behave.
-----------------------------------------------------------------

 [W] anticipatory construct introduced by "it"

 WARNING: use of intensifier "very"

 [W] wordy construct with "interesting" and verb "to be"
     It makes prose insipid. If possible, discard it altogether

 WARNING: vague word "people"

 [W] restrict the use of vacuous words such as "way"
-----------------------------------------------------------------

 *** sentence is unclear ***

  >> complex structure in the predicate
-----------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: simple noun phrase (np)
        comp: adjectival phrase
        comp: complex >> infinitive with complex complement
-----------------------------------------------------------------
```

Example B.17

```
-----------------------------------------------------------------

ANALYSIS OF:  It is interesting to show that they behave selfishly.

-----------------------------------------------------------------

 [W] anticipatory construct introduced by "it"

 [W] wordy construct with "interesting" and verb "to be"
     It makes prose insipid. If possible, discard it altogether
-----------------------------------------------------------------

 <=> SENTENCE IS CLEAR: monoschematic
-----------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: simple noun phrase (np)
        comp: adjectival phrase
        comp: infinitive with embedded clause
-----------------------------------------------------------------
```

Example B.18

```
-------------------------------------------------------------------

ANALYSIS OF:  Like I said, it was to provide an objection.

-------------------------------------------------------------------

 [W]  anticipatory construct introduced by "it"

 WARNING: modifying clause begins with conjunction "like"

 WARNING: inadequate infinitive construct for BE complement

-------------------------------------------------------------------

 <=> SENTENCE IS CLEAR: centroschematic

-------------------------------------------------------------------

 <=> STRUCTURE: initial modifying component + clause

    initial modC -->
        -- subordinate clause --
              clause -->
                  subj: simple noun phrase (np)

    clause -->
        subj: simple noun phrase (np)
        comp: infinitive clause
-------------------------------------------------------------------
```

Example B.19

```
-------------------------------------------------------------------

ANALYSIS OF:  Thou shouldst give me a purple-coloured stamp.

-------------------------------------------------------------------

  USAGE--> archaic form "thou shouldst"

 [W] redundant category: "purple" is a "colour"
     Unnecessary to restate an idea included in a word by definition

-------------------------------------------------------------------

 <=> SENTENCE IS CLEAR: monoschematic

-------------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: simple noun phrase (np)
        comp: simple noun phrase (np)
        comp: simple noun phrase (np)
-------------------------------------------------------------------
```

Example B.20

```
------------------------------------------------------------------

ANALYSIS OF:  My brother says that a computer keyboard is
              rectangular in shape.

------------------------------------------------------------------

 [W] redundant category: "rectangular" is a "shape"
     Unnecessary to restate an idea included in a word by definition

------------------------------------------------------------------

 <=> SENTENCE IS CLEAR: monoschematic

------------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: simple noun phrase (np)
        comp: -- relative clause --
              clause -->
                  subj: simple noun phrase (np)
                  comp: adjectival phrase with prepositional phrase

------------------------------------------------------------------
```

Example B.21

```
------------------------------------------------------------------

ANALYSIS OF:  To fully understand this problem is not impossible.

------------------------------------------------------------------

  *  infinitive verb split by adverb "fully" weakening the
     sentence. A better approach is to put the adverb at the end
     of the infinitive phrase.

 WARNING: double negation: "not" and "impossible"
      negating a negative term in order to convey a qualified
      positive statement is legitimate. It can be effective on
      occasion, but it must not be overworked.

------------------------------------------------------------------

 <=> SENTENCE IS CLEAR: monoschematic

------------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: infinitive clause
        comp: adjectival phrase
------------------------------------------------------------------
```

Example B.22

```
--------------------------------------------------------------------

ANALYSIS OF:  The computer power energy need of this lab is
              tremendous.

--------------------------------------------------------------------

   * excessive noun modification

  USAGE--> colloquial expression "lab"
--------------------------------------------------------------------

 <=> SENTENCE IS CLEAR: monoschematic

       But it would be clearer without the
       excessive number of noun modifiers
--------------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: np with excessive modification or qualification
        comp: adjectival phrase
--------------------------------------------------------------------
```

Example B.23

```
--------------------------------------------------------------------

ANALYSIS OF:  The book that was bought by John is written on
              paper that was recycled.

--------------------------------------------------------------------

  *  3 passives detected

 [W] wordy passive construct in relative clause
     Adjectival wordiness occurs when the writer uses a relative
     clause to introduce a participle that could be attached to
     the noun directly. Occasionally, clarity and emphasis justify
     writing out the entire clause, but in most cases, it is
     better to simply drop the relative pronoun and be auxiliary.
--------------------------------------------------------------------

 <=> SENTENCE IS CLEAR: monoschematic

       But it would be clearer without the
       excessive number of passive verbs
--------------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: np with relative clause
        mod: pp with relative clause
--------------------------------------------------------------------
```

Example B.24

```
------------------------------------------------------------------

ANALYSIS OF:  I left the scene of the accident, shocked, dizzy and
              white as snow.

------------------------------------------------------------------

  *  excessive qualification

  USAGE--> "white as snow" is a cliche

 [W] potential wordiness in the construct X of Y

------------------------------------------------------------------

 *** sentence is unclear ***

  >> discordant element in final position

------------------------------------------------------------------

 <=> STRUCTURE: clause + final modifying component

    clause -->
        subj: simple noun phrase (np)
        comp: np with simple prepositional phrase (pp)


    final modC -->
        complex >> adjectival phrase with excessive qualification

------------------------------------------------------------------
```

Example B.25

```
------------------------------------------------------------------

ANALYSIS OF:  The growth is evident after some time.

------------------------------------------------------------------

 <=> GOOD: stylistically correct

------------------------------------------------------------------

 <=> SENTENCE IS CLEAR: monoschematic

------------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: simple noun phrase (np)
        comp: adjectival phrase with prepositional phrase

------------------------------------------------------------------
```

Example B.26

```
-----------------------------------------------------------------

ANALYSIS OF:  So, I left her with the thought that she was happy
              in the field of medicine.

-----------------------------------------------------------------

  *  as a sentence opener, "so" is colloquial -- Better
     connectives are: "consequently", "therefore", "as a result"

 [W] wordy construct consisting of the abstract noun "thought"
     followed by a relative clause. Consider replacing the
     abstraction with a gerund derived from "thought"

 [W] the "field of" is a weak construct which says
     nothing. It is better avoided
-----------------------------------------------------------------

 <=> SENTENCE IS CLEAR: monoschematic
-----------------------------------------------------------------

 <=> STRUCTURE: initial modifying component + clause

    initial modC -->
        adverb

    clause -->
        subj: simple noun phrase (np)
        comp: simple noun phrase (np)
        mod: embedded clause
-----------------------------------------------------------------
```

Example B.27

```
-----------------------------------------------------------------

ANALYSIS OF:  I left her thinking that she was happy in medicine.

-----------------------------------------------------------------

 <=> GOOD: stylistically correct

-----------------------------------------------------------------

 <=> SENTENCE IS CLEAR: monoschematic

-----------------------------------------------------------------

 <=> STRUCTURE: simple sentence

    clause -->
        subj: simple noun phrase (np)
        comp: np with participial clause
-----------------------------------------------------------------
```

Example B.28

```
------------------------------------------------------------------

ANALYSIS OF:  The engineers still have to get a buffer but they
              are about done.

------------------------------------------------------------------

First Clause:
--------------

  USAGE--> using the verb  "to get" is often colloquial

  USAGE--> "buffer" is a jargon term related to computers

Second Clause:
---------------

  *  1 passive detected

 WARNING: vague construct when using "do" as the main verb of
          a clause

  USAGE--> it is colloquial to use "about" as an adverb

------------------------------------------------------------------

 <=> SENTENCE IS CLEAR: compound structure

------------------------------------------------------------------

 <=> STRUCTURE: compound of sentences

    first sentence:
        clause -->
            subj: simple noun phrase (np)
            comp: infinitive clause

    second sentence:
        clause -->
            subj: simple noun phrase (np)

------------------------------------------------------------------
```

Example B.29

```
-----------------------------------------------------------------

ANALYSIS OF:  It felt like my world had collapsed and it was
              just terrible.

-----------------------------------------------------------------

 [W] same subject is repeated in both sentences

First Clause:
--------------

  USAGE --> introducing a verb complement clause with "like" or
            "like if" is incorrect. Use "as if" or "as though"

Second Clause:
---------------

 [W] anticipatory construct introduced by "it"

  USAGE--> it is colloquial to use the adjective "just" as
           an adverb.

-----------------------------------------------------------------

 <=> SENTENCE IS CLEAR: compound structure

-----------------------------------------------------------------

 <=> STRUCTURE: compound of sentences

    first sentence:
        clause -->
            subj: simple noun phrase (np)
            comp: -- relative clause --
              clause -->
                  subj: simple noun phrase (np)

    second sentence:
        clause -->
            subj: simple noun phrase (np)
            comp: adjectival phrase

-----------------------------------------------------------------
```

Example B.30

References

[Bai88] Alan Bailin. "Artificial Intelligence and Computer-Assisted Language Instruction: A Perspective". *CALICO*, 5(3):22–45, March 1988.

[Bak73] Sheridan Baker. *The Practical Stylist.* Thomas Y. Crowell, 3rd edition, 1973.

[BJMG89] Barbara J. Brecht, Marlene Jones, Gordon I. McCalla, and Jim E. Greer. "Planning the Content for Instruction". In *Proceedings of the 4th International Conference on Artificial Intelligence and Education*, Amsterdam, May 1989.

[BL89] Alan Bailin and Lori Levin. "Introduction: Intelligent Computer-Assisted Language Instruction". *Computers and the Humanities*, 23(1):3–11, January 1989. Special Issue on Intelligent Computer-Assisted Language Instruction.

[Bre91] Barbara J. Brecht. "A Computational Framework for a Content Planner". to appear in: NATO Advanced Research Workshop on Instructional Models in Computer-Based Learning Environments, 1991.

[BT88] Alan Bailin and Philip Thomson. "The Use of Natural Language Processing in Computer-Assisted Language Instruction". *Computers and the Humanities*, 22(2):99–110, 1988.

[Bur88] Jack Burston. "Towards Better Tutorial CALL: A Matter of Intelligent Control". *CALICO*, 6(4):75–89, March 1988.

[Bur90] Gordon Burgess. "Time for Evolution, Time for Change: CALL Past, Present and Future". *Computer Assisted Language Learning*, 1(1):11–18, 1990.

[Car85] Jaime G. Carbonell. "The Role of User Modelling in Natural Language Interface Design". In Stephen J. Andriole, editor, *Applications in Artificial Intelligence*, pages 213–226. Petrocelli Books, 1985.

[Cat88] Mark Edward Catt. "Intelligent Diagnosis of Ungrammaticality in Computer-Assisted Language Instruction". Master's thesis, Department of Computer Science, University of Toronto, October 1988. Published as technical report CSRI-218, Computer Systems Research Institute.

[Cha89] Carol Chapelle. "Using Intelligent Computer-Assisted Language Learning". *Computers and the Humanities*, 23(1):59–70, January 1989. Special Issue on Intelligent Computer-Assisted Language Instruction.

[CJ86] Robin Cohen and Marlene Jones. "Incorporating User Models into Expert Systems for Educational Diagnosis". Research Report CS-86-37, Faculty of Mathematics, University of Waterloo, September 1986.

[Cla88] John L. Clark. "Toward a Research and Development Strategy for Computer-Assisted Language Learning". *CALICO*, 5(3):5–22, March 1988.

[Cor86] Edward P.J. Corbett. "Teaching Style". In Donald A. McQuade, editor, *The Territory of Language: Linguistics, Stylistics, and the Teaching of Composition*, pages 23–33. Southern Illinois University Press, 1986.

[CV86] L.L. Cherry and W. Vesterman. "Writing Tools - The STYLE and DICTION Programs". In *Unix User's Manual: Supplementary Documents*, chapter 32. University of California, Berkeley, BSD 4.3 edition, 1986.

[Dav82] Graham Davies. *Computers, Language and Language Learning.* Center for Information on Language Teaching and Research, 1982.

[DiM89] Chrysanne DiMarco. *Computational Stylistics for Natural Language Translation.* PhD thesis, Department of Computer Science, University of Toronto, December 1989. Published as technical report CSRI-239, Computer Systems Research Institute.

[Dob90] David N. Dobrin. "A New Grammar Checker". *Computers and the Humanities*, 24(1):67–80, 1990.

[Duc88] P.C. Duchastel. "Models for AI in Education and Training". In P. Ercoli and R. Lewis, editors, *Artificial Intelligence Tools in Education*, pages 17–28. North-Holland, 1988. Proceedings of the IFIP TC3 Working Conference on Artificial Intelligence Tools in Education, May 1987.

[Eas70] Richard M. Eastman. *Style: Writing as the Discovery of Outlook.* Oxford University Press, 1970.

[EL89] David A. Evans and Lori S. Levin. "Toward the Design of Natural-Language Instructional Systems: Notes from ALICE". Department of Computer Science, Carnegie Mellon University, 1989.

[Far89a] Ali Farghaly. "A Model for Intelligent Computer Assisted Language Instruction (MICALI)". *Computers and the Humanities*, 23(3):235–250, June 1989.

[Far89b] Brian Farrington. "AI: Grandeur or Servitude?". In Keith Cameron, editor, *Computer Assisted Language Learning*, pages 67–80. Ablex Publishing, 1989.

[Fer89] Derrik Ferney. "Small Programs that 'Know' What They Teach". In Keith Cameron, editor, *Computer Assisted Language Learning*, pages 14–27. Ablex Publishing, 1989.

[Fow61] Henry Watson Fowler. *A Dictionary of Modern English Usage.* Oxford University Press, 2nd edition, 1961.

[Fra86] Marcella Frank. *Modern English: Exercises for Non-native Speakers.* Prentice Hall, 2nd edition, 1986. Part 1: Parts of Speech; Part 2: Sentences and Complex Structures.

[GB89] J.E. Galletly and C.W. Butcher. "Towards an Intelligent Syntax Checker". In Keith Cameron, editor, *Computer Assisted Language Learning*, pages 81–100. Ablex Publishing, 1989.

[Gri75] H.P. Grice. "Logic and Conversation". In Peter Cole and Jerry L. Morgan, editors, *Syntax and Semantics*, volume 3. Academic Press, 1975.

[Hai87] Douglas Hainline, editor. *New Developments in Computer-Assisted Language Learning*. Nichols Publishing, 1987.

[Ham90] Henry Hamburger. "Evaluation of L2 Systems: Learners and Theory". *Computer Assisted Language Learning*, 1(1):19–27, 1990.

[Har83] Jeremy Harmer. *The Practice of English Language Teaching*. Longman Handbooks for Language Teachers. Longman, 1983.

[Hir87] Graeme Hirst. *Semantic Interpretation Against Ambiguity*. Cambridge University Press, 1987.

[HJM+82] G.E. Heidorn, K. Jensen, L.A. Miller, R.J. Byrd, and M.S. Chodorow. "The EPISTLE Text-critiquing System". *IBM Systems Journal*, 21(3):305–326, 1982.

[HL82] James A.W. Hefferman and John E. Lincoln. *Writing. A College Handbook*. W.W. Norton & Company, 1982.

[Hov87] Eduard H. Hovy. *Generating Natural Language under Pragmatic Constraints*. PhD thesis, Yale University, May 1987. Published as research report 521, Department of Computer Science.

[Iri89] Estelle Irizarry. "Exploring Conscious Imitation of Style with Ready-made Software". *Computers and the Humanities*, 23(3):227–233, June 1989.

[Jon86] Stephen Gwyn Jones. "Language, Teaching, Communication and IKBS". In K.S. Gill, editor, *Artificial Intelligence for Society*, pages 245–251. John Wiley, 1986.

[KA88] Istvan Kecskes and Laszlo Agocs. "Expressing the Same Piece of Information Using Different Elements of Language: A Text-Based System". *CALICO*, 7(1):53–65, March 1988.

[Kan83] Thomas S. Kane. *The Oxford Guide to Writing: A Rhetoric and Handbook for College Students*. Oxford University Press, 1983.

[KF88] Robert Kass and Tim Finin. "Modeling the User in Natural Language Systems". *Computational Linguistics*, 14(3):5–21, September 1988. Special Issue on User Modeling.

[KK83] M.J. Kenning and M-M. Kenning. *An Introduction to Computer Assisted Language Learning*. Oxford University Press, 1983.

[KW88] Alfred Kobsa and Wolfgang Wahlster, editors. *User Models in Dialog Systems*. Springer Verlag, 1988.

[Lan74] Richard A. Lanham. *Style: An Anti-Textbook*. Yale University Press, 1974.

[Las89] R.W. Last. *Artificial Intelligence Techniques in Language Learning*. Ellis Horwood Series in Computers and their Applications. Halsted Press, a division of Wiley, 1989.

[l'H90] Monique l'Huillier. "Evaluation of CALL Programs for Grammar". *Computer Assisted Language Learning*, 1(1):79–86, 1990.

[MFK82] N.H. Macdonald, G.P. Frase, and S.A. Keenan. "The WRITER'S WORKBENCH: Computer Aids for Text Analysis". *IEEE Transactions on Communication*, 30(1):105–109, 1982.

[Mil86] Louis T. Milic. "Composition via Stylistics". In Donald A. McQuade, editor, *The Territory of Language: Linguistics, Stylistics, and the Teaching of Composition*, pages 192–203. Southern Illinois University Press, 1986.

[Mul89] George W. Mulford. "Semantic Processing for Communicative Exercises in Foreign Language Learning". *Computers and the Humanities*, 23(1):31–44, January 1989. Special Issue on Intelligent Computer-Assisted Language Instruction.

[Neu89] Christine M. Neuwirth. "Intelligent Tutoring Systems: Exploring Issues in Learning and Teaching Writing". *Computers and the Humanities*, 23(1):45–57, January 1989. Special Issue on Intelligent Computer-Assisted Language Instruction.

[Ng84] Evelyn Kam Ling Ng. "Computer-Assisted Language Learning: Some Design and Implementation Considerations". Master's thesis, Faculty of Education, University of Toronto, 1984.

[NG88] Sarah Norton and Brian Green. *The Bare Essentials: Form B.* English Writing Skills Series. Holt, Rinehart and Winston of Canada, 2nd edition, 1988.

[Opp88] Rosa Oppenheim. "The Mathematical Analysis of Style: A Correlation-Based Approach". *Computers and the Humanities*, 22(4):241–252, 1988.

[OS83] Tim O'Shea and John Self. *Learning and Teaching with Computers: Artificial Intelligence in Education.* The Harvester Press, 1983.

[Par88] Cécile L. Paris. "Tailoring Objects Descriptions to a User's Level of Expertise". *Computational Linguistics*, 14(3):64–78, September 1988. Special Issue on User Modeling.

[PS87] Fernando C.N. Pereira and Stuart M. Shieber. *Prolog and Naturla Language Analysis.* Center for the Study of Language and Information, Stanford University, 1987. CLSI lecture notes 10.

[RBH88] Stephen D. Richardson and Lisa C. Braden-Harder. "The Experience of Developing a Large-Scale Natural Language Text Processing System: CRITIQUE". In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 195–202, Austin, Texas, February 1988.

[RR65] Paul I. Richards and Irving T. Richards. *Proper Words in Proper Places.* The Christopher Publishing House, 1965.

[SB82] D. Sleeman and J.S. Brown, editors. *Intelligent Tutoring Systems.* Computers and People Series. Academic Press, 1982.

[Sch86] Sandra Schor. "Style through Control: The Pleasures of the Beginning Writer". In Donald A. McQuade, editor, *The Territory of Language: Linguistics, Stylistics, and the Teaching of Composition*, pages 204–212. Southern Illinois University Press, 1986.

[Sel87] John Self. "Student Models: What Use Are They?". In P. Ercoli and R. Lewis, editors, *Artificial Intelligence Tools in Education*, pages 73–86. North-Holland, 1987. Proceedings of the IFIP TC3 Working Conference on Artificial Intelligence Tools in Education, May 1987.

[Sel88] John Self, editor. *Artificial Intelligence and Human Learning: Intelligent Computer-Aided Learning.* Chapman and Hall, 1988.

[Und89] John Underwood. "On the Edge: Intelligent CALL in the 1990's". *Computers and the Humanities*, 23(1):71–84, January 1989. Special Issue on Intelligent Computer-Assisted Language Instruction.

[Vis88] A. Visalberghi. "Artificial Intelligence: Educational Problem and Implications". In P. Ercoli and R. Lewis, editors, *Artificial Intelligence Tools in Education*, pages 1–4. North-Holland, 1988. Proceedings of the IFIP TC3 Working Conference on Artificial Intelligence Tools in Education, May 1987.

[Wie81] Harvey S. Wiener. *The Writing Room: A Resource Book for Teachers of English.* Oxford University Press, 1981.

[Wil81] Joseph M. Williams. *Style: Ten Lessons in Clarity and Grace.* Scott Foresman and Company, 1981.

[Win84] T. Winograd. "Computer Software for Working with Language". *Scientific American*, 8(9):131–145, September 1984.

[WVJ78] R.M. Weischedel, W.H. Voge, and M. James. "An Artificial Intelligence Approach to Language Instruction". *Artificial Intelligence*, 10(3):225–240, November 1978.

[Yaz89] Masoud Yazdani. "Language Teaching with PROLOG". In Keith Cameron, editor, *Computer Assisted Language Learning*, pages 101–111. Ablex Publishing, 1989.

[ZLF89] M. Zock, A. Laroui, and G. Francopoulo. "SWIM: A "Natural" Interface for the Scientifically Minded Language Learner". *Computers and the Humanities*, 23(4-5):411–422, August-October 1989. Proceedings of the 8th international conference on Computer and the Humanities.