Heift, Trude and Mathias Schulze (2007) *Errors and Intelligence in Computer-Assisted Language Learning. Parsers and Pedagogues.* New York: Routledge.

(below is a draft of the table of contents and the introduction to the book)

Table of Contents

Introduction

> I would like to offer a test for the seriousness of the claim that a computer has consciousness. I will only believe someone who is willing to give a present to a computer. If s/he answers: This is impossible; I would reply: Indeed. The computer does not have a world, but only information which it processes on the basis of programs and this it can do much better than we can. We, on the other hand, can be shocked when we realize that we only follow a program automatically without any purpose and meaning and without any consideration for the world in which we live. (R. Schröder, 1998)[1]

1.1. Intelligence and Parsers

The concept of intelligence as applied to Computer-Assisted Language Learning (CALL[2]) has been a source of contention since the early 1980s. The contribution that Artificial Intelligence research and techniques can make and have made to CALL has been exaggerated (O'Brien, 1993), misunderstood (Last, 1989) and doubted (R. Salaberry, 1996). At the same time, areas of Artificial Intelligence which are relevant to CALL, such as Natural Language Processing (NLP), Student Modeling (SM) and Intelligent Language Tutoring Systems (ILTSs) have played a significant role in the development of our thinking about CALL, its design and implementation. For instance, Nerbonne (2003), in his chapter on NLP in CALL in the Oxford Handbook of Computational Linguistics, argues that recent advances in NLP have much to contribute to CALL. Borin et al. (2003) put forward a similar argument in their proposal to the European Union to establish a Network of Excellence (Supporting Second Language Acquisition with Human Language Technology (SLAinTe)). They ascertain that there is a need for research in this area because recent research in language technology has had little impact on approaches in CALL.

Marshall (1988) identifies the significant interactive qualities of CALL as one advantage of using the computer in the language classroom. True interaction, however, requires intelligent behavior on the part of the computer. Without intelligence, the system is merely another method of presenting information, one not necessarily preferable to a static medium like print. Instead of multiple choice questions, relatively uninformative answer keys and gross mainstreaming of students – characteristic of workbooks, NLP-based CALL is aiming at interactive computer systems possessing a high degree of artificial intelligence and capable of processing natural language input (Holland *et al.*,

---

[1] Translated from the German original: "Ich will einen Test anbieten auf die Ernsthaftigkeit der Behauptung, daß ein Computer Bewußtsein hat. Das glaube ich nur dem jenigen, der bereit ist, *einem Computer etwas zu schenken*. Wenn er antwortet: Das geht doch gar nicht, würde ich entgegnen: Eben. Der Computer hat keine Welt, sondern bloß Informationen, die er nach Programm verarbeitet, und dies kann er besser als wir. Wir aber können außerdem über uns erschrecken, wenn wir bemerken, daß wir bloß noch automatisch, ohne Sinn und Verstand und ohne Rücksicht auf die Welt, in der wir leben, ein Programm abarbeiten."

2 The acronym CALL will be used throughout this book. A number of synonymous acronyms have been used in the literature (see Levy, 1997, pp. 77-79 for an overview).

1993). The strength of NLP is that it allows for a sophisticated error analysis where student tasks can go beyond multiple-choice questions and/or fill-in-the-blanks. Many simple drills, on the other hand, are usually based on string matching algorithms, that is, the student response is compared letter for letter against an answer key. However, one obviously cannot store the infinitely many sentences required for meaningful practice for purposes of comparison. NLP provides the analytical complexity underpinning an ILTS.

Matthews (1993) identifies Linguistic Theory and Second Language Acquisition (SLA) Theory as the two main disciplines which inform and have been informed by Intelligent Computer-Assisted Language Learning (ICALL). He adds that "the obvious AI research areas from which ICALL should be able to draw the most insights are Natural Language Processing (NLP) and Intelligent Tutoring Systems (ITSs)" (1993, p. 5). Matthews also shows that it is possible to "conceive of an ICALL system in terms of the classical ITS architecture" (Matthews, 1993, p. 6). An ITS generally consists of three modules: An expert, a student and a teacher module.[3] The expert module houses the language knowledge and, ideally, it is this part that can process any piece of text produced by a language learner. This is usually achieved by a parser. A parser produces a formal linguistic representation of natural language input by identifying the grammatical functions of the parts of a sentence[4]:

> The use of parsers in CALL is commonly referred to as intelligent CALL or 'ICALL', it might be more accurately described as parser-based CALL, because its 'intelligence' lies in the use of parsing – a technique that enables the computer to encode complex grammatical knowledge such as humans use to assemble sentences, recognize errors and make corrections. (Holland et al., 1993, p. 28)

This notion of parser-based CALL not only captures the nature of the field much better than the somewhat misleading term *Intelligent CALL* (Is all other CALL *un-intelligent*?), but it also identifies the use of human language technology as one possible approach in CALL alongside others, such as multimedia-based CALL and web-based CALL. In some cases, the (technology-defined) borders between these sub-fields of CALL are not even clearly identifiable as illustrated in some of the projects described later. For this reason, the terms *parser-based systems*, *ICALL systems* and *Intelligent Language Tutoring Systems* (ILTSs) are used interchangeably throughout this book although, for the reasons outlined above, the term parser-based systems or parser-based CALL are preferred.

The following quote illustrates that the notion of intelligence is perceived in a number of different ways in CALL. Rüschoff (1987) contrasted two of these notions and concluded his discussion of *intelligence* in CALL as follows:

> Returning, however, to the question of how "intelligent" computer assisted language learning materials have to be in order to make them suitable for effective self-study use, the solution to this problem does not necessarily lie in the development of true expert systems or any other form of artificial intelligence. Such systems are still very much a thing of the future and we shall have to investigate

---

[3] In addition, the system also contains an interaction module.
[4] Parsers and parsing will be discussed in some detail in part 2.

> their possible points of application for computer assisted language learning materials when they become available. However, I do hope we do not have to wait until then to develop useful CALL materials for self-study. Such programs would not need an "intelligence" of their own, but their quality would rather depend on the intelligence with which they have been programmed. (pp. 81-82)

This is certainly true, but at the same time, it does not suggest that a parser-driven CALL application will not prove useful in language learning. Examples throughout this book will illustrate that parsing technology can provide CALL with significant advantages over traditional workbook instruction. For instance, studies (e.g., Heift, 2001, 2002, 2003, 2004; Nagata, 1996, 1998a, 1998b; van der Linden, 1993) addressing the question of what kind of feedback a computer program should give have shown that not only do students appreciate the more sophisticated, error-specific feedback made possible by NLP, but also perform better on the language skills being taught. The fact that students learn better provides the rationale for employing parsers in CALL. Ideally, the advantages of a grammar-checking component in CALL software are complemented by intelligent communicative tasks that facilitate effective language learning.

Higgins (1987), in his article entitled "Artificial Unintelligence: Computer Uses in Language Learning", puts forward an interesting notion of the term *intelligence* in CALL:

> Yet there is intelligence present during the interaction [students using software programs produced by Higgins] – the learner's intelligence in assessing, responding to, criticizing, or enjoying what the machine sets up – and it seems that the learner's recognition of the machine's stupidity is a factor in releasing their own intelligence and zest for experiment. (p. 162)

Most misgivings identified in the CALL community with the term *Intelligent CALL* can be put down to two reasons:

1. The term intelligence has been transferred from the area of Artificial Intelligence (AI) without making its background in this discipline transparent in CALL. For instance, for many language teachers, *intelligence* by its very definition is an inherently human category. In contrast, the notion of intelligence in computing has been defined with computers in mind (e.g., a test proposed by Alan Turing, in which a (conversation) machine is said to have intelligence if a human interrogator cannot distinguish it from a human (Brookshear, 1997, p. 359).
2. The use of *intelligent* for programs that can be said to approximate the passing of the Turing test appears to be common practice which probably led to the coining of the term Intelligent CALL. A closer linguistic analysis of the terminology reveals another problem, that of scope ambiguity of the modifier *intelligent*. The modifier is adjacent to the noun computer which is part of the compound *computer-assisted*. If *intelligent* were to refer to computer then its use would be justified because of the artificial intelligence nature of the programs to which the whole term refers. The problem is that *intelligent* is much more likely to be interpreted as the modifier of the head noun *learning* and thus it stipulates that only CALL that relies on parsers is intelligent. Undoubtedly, this notion cannot have been intended by the proponents of the use of artificial intelligence

The notion is certainly not intended by us in this book in which we will discuss different aspects of artificial intelligence in CALL in some detail.

1.2. The Structure of the Book

The use of NLP techniques and tools in CALL has indeed become one of the major areas of research and development in the field of new technologies and language learning. An increasing proportion of articles in journals dedicated to CALL (e.g., CALL, ReCALL, CALICO) are devoted to the discussion of NLP techniques relevant to the field. However, there is no book that gives a comprehensive overview of the research issues involved and that offers a solid introduction for students and researchers working in CALL, Applied Linguistics and/or Computational Linguistics. Unlike recent publications (e.g. Heift & Schulze, 2003b; Holland *et al.*, 1995; Jager *et al.*, 1998; Schulze *et al.*, 1999; Swartz & Yazdani, 1992) which are compilations of papers usually derived from a conference, this book provides an overview of theoretical issues, historical developments and current trends and gives due attention to all of these areas in a balanced way. It assumes a basic familiarity with Second Language Acquisition (SLA) theory and teaching, CALL and linguistics. The book is written for upper undergraduate and/or graduate students who study CALL, SLA, Language Pedagogy, Computational Linguistics or Artificial Intelligence as well as researchers with a background in any of these fields.

The book consists of four main parts. Starting off with a short review of (artificial) intelligence in CALL and describing the role it has played and how it has been discussed, the authors offer an overview of research and development at the intersection of NLP and CALL. This overview introduces key concepts in the area of parser-based CALL systems by discussing individual projects. Moreover, the reader gains an understanding of grammar formalisms and parsing algorithms and their relevance to language learning software. Relevant literature from neighboring disciplines that have provided insights into error diagnosis and feedback are reviewed and contextualized within the area of parser-based CALL. Error diagnosis and feedback are two processes in language learning that are of importance to the research, development and implementation of ICALL systems. Both concepts are investigated in considerable detail to discuss current trends in research and development and related research in SLA theory. The final part provides an overview of language learning systems that make use of student modeling techniques to achieve a more individualized language learning environment. Student modeling techniques that have been applied to both NLP-based systems and other CALL programs will be presented. A concluding chapter considers future directions including empirical research.

The following sections provide a more detailed description of each part of the book.

1.2.1. Part 2: NLP in CALL

Part 2 introduces the reader to important concepts in natural language understanding and processing and aims at familiarizing the reader with different approaches of NLP

techniques in CALL, its success stories and failures. The notion of a computational grammar will be discussed and different parsing techniques will be introduced. For instance, parsers designed for language instruction typically contain components that anticipate or search for errors in the event that a grammatical rule is not successful, buggy rules being a common instance. Lately, researchers have also analyzed L2 corpora to determine certain buggy rules that are needed for a language parsing system. In addition, statistical methods have also been applied to error detection and diagnosis.

Selected research and development projects of the past 25 years will be presented in chronological order to introduce key concepts, such as different parsing techniques, programming and natural languages, and to show the development of the field.

1.2.2. Part 3: Error Analysis and Description

Part 3 begins with an overview of the inherent challenges of designing spell and/or grammar checkers. It then continues with a discussion of Error Analysis, a branch of Applied Linguistics that influenced our understanding of Second Language Acquisition (SLA) in the late 1960s and early 1970s. The advent of electronic learner corpora and the need for elaborate error data has sparked a recent renewed interest in the methodologies employed by Error Analysis (EA). We will show how findings in EA can be successfully employed in parser-based and other CALL systems. In the final part, corpus studies will be discussed in some detail. Their role in our understanding of language learning, in general and CALL, in particular will be presented.

1.2.3. Part 4: Feedback

Part 4 discusses the notion of feedback in language learning and CALL under five different aspects: human-computer interaction (HCI), learning theories, SLA, formal grammar and, of course, CALL.

Human-computer interaction (HCI), as an area of research generally deals with questions of interface and dialogue design. The discussion here focuses on interface design. Findings of HCI research that are applicable to the provision of feedback by parser-based software for foreign-language learner will be presented. The concept of reinforcement is frequently employed in behaviorist learning psychology and is related to the concept of feedback. The assumption is that feedback of a certain kind can act as a reinforcer. Socio-cultural concepts in psychology form the basis for our discussion and relevant phenomena and findings of behaviorist and cognitive psychology will be outlined. SLA research that is relevant to feedback and the design of parser-based CALL systems will also be reviewed at this stage. Formal Linguistics provides the tools to describe human language and thus learner language in a systematic and mathematical way with the goal for it to be *understood* by a computer. We will discuss the representation and connection of well-formed and ill-formed sentences. This part will then conclude with an overview of related topics, e.g., design issues, and research on feedback in CALL.

1.2.4. Part 5: Student Modeling

Part 5 discusses the theoretical issues surrounding individualized instruction and it provides examples of parser-based and other CALL systems that have implemented

different kinds of student models and modeling techniques. Individualized language instruction has long been recognized as a significant advantage of CALL over workbook tasks. A *one size fits all* approach is not appropriate for a learning environment. Students learn at their own pace and often, work for their own purposes. Learners also vary with respect to prior language experience, aptitude, cognitive needs, beliefs, attitudes and/or learning styles and strategies.

Student models are challenging in a number of ways. Self (1979), for example, in opposition to a number of other researchers who list the advantages of student models, discusses the difficulties of how to capture what kind of information and how to maintain and implement it. Such difficulties have since been the subject of some discussion and we will provide glimpses of these discussions and report steps towards solving these problems.

1.2.5. Part 6: The Past and the Future

Part 6 provides concluding remarks that consider the past 25 years and future directions of NLP in CALL. We look at different ways of identifying current trends and future developments and make an attempt to apply these to the field of NLP in CALL. We will pay particular attention to the need for future empirical studies.