Conceptual Modelling in Error Analysis in Computer-Assisted Language Learning Systems

Thierry Chanier, Michael Pengelly, Michael Twidale, John Self

▶ To cite this version:

HAL Id: edutice-00000279

https://edutice.archives-ouvertes.fr/edutice-00000279v2

Submitted on 25 Feb 2012

Conceptual Modelling in Error Analysis in Computer-Assisted Language Learning Systems

Thierry Chanier[1], Michael Pengelly[2], Michael Twidale[2], and John Self[2]

[1] ICF (Research Assoc.: Ingéniérie Cognitive et Formation), 49 av Jean Jaurès, 63400 Chamalières, France.
[2] Computing Department, Lancaster University, Lancaster, LA1 4YR, UK

Abstract: Many computer-assisted language learning systems specifically designed to be used in the curriculum and which exploit AI techniques have neither a learner model nor consequently any deep error analysis. Evidence from applied linguistics shows that learners have their own system of rules for the production of a foreign language. We believe the central issue is to determine the appropriate level of description of these rules and uncover the strategies used by the learners in particular situations. This information represents the major part of the learner model. We review error analysis in second language learning and tutoring systems related to this perspective. We introduce a new structure, called an "applicable rule", that can be used to help diagnose and to represent a learner's performance. We propose a design for the architecture of a system for computer diagnoses of learners' grammatical performances in a communicative environment. Examples of diagnosis using applicable rules illustrate the functioning of this architecture.

Keywords: Learner modelling, Error Diagnosis, Intelligent Tutoring System.

1. Introduction

Evidence in the literature of applied linguistics indicates that learners have their own systems of rules for the production of the foreign language that they practise. It suggests that they use various heuristics and strategies both to *acquire* these rules and to *use* them in communicative processes. We would like to find an appropriate description of these rules and determine what strategies are being used in a given situation to diagnose better a learner's performance as a major contribution towards the learner model, a primary element of an intelligent tutoring system (ITS).

Work in the field of applied linguistics suggests potential methods of modelling learners' behaviours in terms of their strategies of language acquisition. These methods have found little application in building computer-assisted language learning (CALL) systems, which when employing learner modelling techniques still view errors as undesirable elements of performance and something to be prevented and have tended only to diagnose the learner's behaviour at a superficial level, thus following the traditional mal-rule/bug catalogue approach of ITS. Many explanations account for this situation: there is a lack of formalised human expertise on the subject; the management of non-strictly predefined divergences is a hard computational problem which lies on the edge of the main stream of computational linguistics; and the harmonisation of a number of knowledge sources, computational linguistics, pedagogical knowledge of grammar, learning and teaching strategies, in a collaborative manner is not an easy task.

In our approach to learner modelling we follow the direction of the applied linguistics community in viewing second language learning (SLL) as a process of building and updating rules for language production through the use of heuristics and strategies brought about by the circumstances in which a learner finds herself. This positive view of the rules a learner is currently employing de-emphasises the learners failures. The process of learner modelling is to identify the set of rules and their causal mechanisms that are currently active. Because this diagnosis requires interacting with the learner and reasoning about competing explanations of performance, an appropriate representation for both activities must be found.

We introduce here the concept of "applicable rules" (AR) as a basis for this representation and illustrate their use in dialogues between a learner and a system, we call BELLOC, that we are developing in order to automatically build a learner model through the diagnosis of her applicable rules. Although the use of the applicable rules is immersed within a learner-system dialogue environment, semantic problems associated with following the dialogue, such as modelling the learner's actions and intentions related to the problem she has to solve [22], are not considered. But the communicative environment within which they are used provides the context for their diagnosis.

The specification of the BELLOC system has been through a three phase approach. Using the available evidence we proposed a tentative structure for an applicable rule that incorporated the knowledge needed for use in automated learner modelling. We then set up two experiments: the first to acquire the knowledge to partially validate this structure; the second to use the rules in a dialogue game with learners to further validate and refine the learner modelling process. An interface for the the first experiment has been developed and has been used to acquire applicable rules from experts [30,11]. The third phase was to propose an architecture for BELLOC and illustrate its functioning, and that is our focus here.

Section 2 introduces the notion of conceptual diagnosis of a learner's rules through a sample dialogue between the learner and the system. In section 3 the procedure of Error Analysis in applied linguistics and its influence on the perception of language acquisition are described. The learner's main strategies for the acquisition of a second language are sketched. Section 4 gives a snapshot of the mal-rule and bug catalogue modelling techniques frequently encountered in ITSs and briefly indicates their shortcomings. Section 5 surveys ITSs for second language learning, pointing out their treatment of errors and their modelling techniques. Section 6 is devoted to the BELLOC system. It presents an architecture, which has not yet been implemented, for use in the diagnosis of learners' applicable rules, defines refined versions of applicable rules, and illustrates their use though examples. Lastly section 7 gives the conclusions and a perspective on future research.

2. Conceptual diagnosis in a communicative environment

In order to give a flavour of what we mean by a conceptual diagnosis of a learner's errors, we first present an imaginary dialogue between a learner and the system in the context of our BELLOC application.

Much SLL ITS work has been restricted to relatively narrow, artificial exercises. To avoid this situation and to have an opportunity of moving towards a more open learning environment we have tried to place the learner in a problem-solving situation where the second language is being used for a purpose, but where the domain of discourse is sufficiently restricted so that a system could be developed to understand the meaning of inputs well

enough to sustain a dialogue. The problem solving situation we propose is for learners to try to determine what inheritance is due to them, given the conditions of a will, by finding out details of past and present family relations.

At the present stage, the dialogue is sustained off-line with English native learners asking their questions in French. Here, for the sake of the presentation we have extracted some pieces of the dialogue just before the occurrence of an error and have imagined how the system would diagnose the learner's rules and maintain a learner model.

In the dialogue[1] of Figure 2.1 the learner has uttered a sentence that includes two errors. When trying to find out the address of Jean's sister, she said: "Que est sa adresse ?" instead of "Quelle est son adresse ?". Leaving aside the problem of tackling two errors at the same time, the system focuses on the diagnosis of the learner's behaviour relating to the mismatch between "sa" and "adresse".

```
Dialogue1 ("L" stands for "learner" and "S" for "system")

L: Jean a-t-il une soeur? (Has Jean a sister ?)
S: Oui.
L: Est-elle vivante ? (Is she alive ?)
S: Oui.
L: Que est sa adresse ? (What is her address ?)
S: There is a problem in your last sentence.
Would you accept this utterance (yes/no) ? :
(Cécile lives with her brother) Cécile vit avec son frère. (i)
L: yes
S: OK. Would you accept this utterance (yes/no) ?:
ta autre soeur (ii)
L: yes
S: I think the problem is the following:
You cannot say: ta autre soeur (iii)
But you can say: ta soeur, ton autre soeur
Explanation (yes/no) ?
L: yes
S: You can also say: ma belle héritière, mon héritière
Because : the masculine forms "mon", "ton", "son" are found
qualifying a feminine singular noun, when they are followed by a
word beginning with a vowel or h-mute.
Do you want a complete presentation of the rule
<Possessive Adjective Agreement> (yes/no) ? (iv)
```

Figure 2.1: Discrimination between two possible explanations of the learner's behaviour relating to a morphological agreement error between "sa" and "adresse".

The system has two possible explanations of this error relating to two different *applicable rules* (definition of an AR is given in section 6). It is going to discriminate between them in order to know which one may be applicable to the learner. First the system tries to discover if this is a transfer problem, i.e. an application of an English grammar rule (gender agreement with the possessor when, in French, the possessive adjective agrees in gender with the possessed object). The system uses a discriminative example associated with the first applicable rule: it asks the learner her opinion about sentence (i). Since she accepts this sentence, it means that the English rule is not applicable, provided she shows this behaviour consistently[2].

---

[1] [1] In the kind of dialogue presented here and in the other examples the interactions with the user are very limited. We give them only to illustrate our concern, not as an example of typical dialogue to hold in an ITS

[2] This may be considered as a strong hypothesis because we often are inconsistent when learning. But in this paper we will restrict ourselves to the diagnosis of some learner's rules consistently applied.

Then the system tries the other explanation: the learner ignored or forgot the morphological agreement between the possessive adjective "sa" and the following noun. The teacher's applicable rule describing this agreement has the discriminative example (ii). Since the learner accepts it, the system interprets the answer as a rejection of this rule, which is a special case of the general rule of the possessive adjective agreement. It deduces that the learner has used a strategy of simplification when learning and thus "forgot" this special case. Having found the explanation of the learner's behaviour, the system can now present the teacher's applicable rule using the examples (iii) and a pedagogical explanation. It can also propose presenting to the user the general rule (iv). After this dialogue the learner's model is updated and contains (among others) the following learner's applicable rule :

```
Learner model:
AR3.1, Learner's rule
Divergent sentence: Que est sa adresse ?
Name: Incomplete application of rule
<AR2,Possessive Adjective Morphological Agreement>
Causal explanation:
Learning strategy, simplification
...
```

In order to perform this diagnosis, different kinds of knowledge and techniques need to be used. We will present them later on and detail the applicable rules which have been referred to in this example. Before we explain our system in detail, we first review Error Analysis and traditional methods of learner modelling in ITS. Then we review other approaches to modelling SLL as background.

3. Error Analysis in Applied Linguistics

First we have to make a clear distinction between what are errors on the one hand and lapses and slips on the other hand. In the Applied Linguistics field, the term "error" is taken to mean some idiosyncratic or 'un-nativelike' piece of language produced regularly and systematically by a foreign language learner. Lapses and slips refer to occasional actions which are not systematic and which the learner herself can correct. They are often called mistakes. They are of no interest to the Error Analysis since they say nothing about the true state of the learner's knowledge.

Error Analysis has a long tradition. Until the late 1960s, it consisted mainly in building large collections of "common" errors and their linguistic classification [16]. The goals were pedagogic: errors provided information for teaching and for designing remedial lessons. No theoretical framework existed for explaining the role played by the errors. In accordance with Behaviourist learning theory, the prevention of errors was more important than the identification of errors. Most people considered errors as a faulty version of the foreign language. The Contrastive Analysis Hypothesis[3] reinforced this view with the notion of "interference": existing habits prevent correct utterances from becoming established; errors are signs of learning failure and thus are not tolerable.

At the beginning of the 1970s new studies [34] changed the approach to the Error Analysis problem. Following the new Mentalist theory of language learning, these theorists asserted that the second language learners could be viewed as actively constructing rules from the data they encounter and gradually adapting these rules in the direction of the target-language

---

[3] According to the Contrastive Analysis, L2 errors are the result of differences between the learner's first language and the foreign language.

system. This means that learners' errors need not be seen as signs of failure. On the contrary, they give some evidence for the learner's development systems [24]. Today theories of language learning are increasingly learner-oriented. Error Analysis is still of interest since errors which represent the product of learning, can be used to get hints about the underlying process of learning and particularly the learner's strategies.

Next we will briefly introduce the general procedure of Error Analysis and examine more closely the step concerned with the explanations. Some learner's strategies will be illustrated before drawing conclusions on the interest of Errors Analysis to ITSs in second language learning.

3.1. What is Error Analysis ?

We will describe the procedure for Error Analysis following the six-steps method set by van Els & al [44]. (1) A corpus of language is selected. This involves deciding on the size of the sample, the medium to be sampled and the homogeneity of the sample (by taking into account the learner's characteristics: age, stage of development, motivation, etc.). (2) The errors in the corpus are identified. Here one needs to distinguish between true errors and slips. Errors can be "overtly idiosyncratic" (ill-formed in terms of target language rules) or "covertly idiosyncratic" (sentences superficially well-formed but when taking the context into account they are clearly ungrammatical). (3) The errors are classified. This involves assigning a linguistic description to each error. (4) The errors are explained. In this stage of the procedure, an attempt is made to identify the psycholinguistic cause of the errors. (5) The errors are evaluated. This stage involves assessing the seriousness of each error in order to make principled teaching decisions which are to be processed in step (6), prevention/correction of errors.

In this paper we are mainly interested by the first four steps. Step one is determined by the choice of the application: in BELLOC the coverage of the corpus is essentially centred around interrogatives in French. The identification of the errors and their location within the linguistic knowledge described in the ITS is a hard problem which will be referred to later on. The classification of an error (step three) in the procedure for Error Analysis consists of characterising the linguistic knowledge concerned by the error.

Some language books present a wide range of typical errors classified by knowledge sources which can give useful hints when developing a system. Swan and Smith [39], for example, presents a good overview of the different sources relating to interference problems between French native speakers speaking English: grammar, phonology, morphology, punctuation, vocabulary. But these criteria are insufficient for classifying an error in an ITS: firstly, because the way the linguistic knowledge is described does not allow a clear distinction between these sources and, secondly, because several different levels of such knowledge are often involved in an error. One needs to distinguish, as we will see, the level of generality of the rules describing the error and the kinds of techniques needed to tackle them. Step four, the explanations of errors, is our main concern here. We will now detail the related applied linguistics approach.

3.2. Learner's processes and explanations of interlanguage errors

One of the most significant contributions of Error Analysis lies in its success in elevating the status of errors from undesirability to that of a guide to the learner's internal processing used

for *acquiring* L2 knowledge and *using* it for creating speech. These processes/strategies are often classified into three different types [16]: strategies of learning (how to internalise and automatise L2 knowledge), of production (how to use existing resources automatically), and of communication (how to compensate for inadequate resources). The first type of process relates to the learning language aspect, the others two types to the use of it, the whole set of cognitive processes provide us with a general explanation of the second language acquisition (or the interlanguage[4]) problem. Whilst presenting these strategies, we associate the respective explanations of errors. Figure 3.1 sums up the links between processes/strategies and errors.
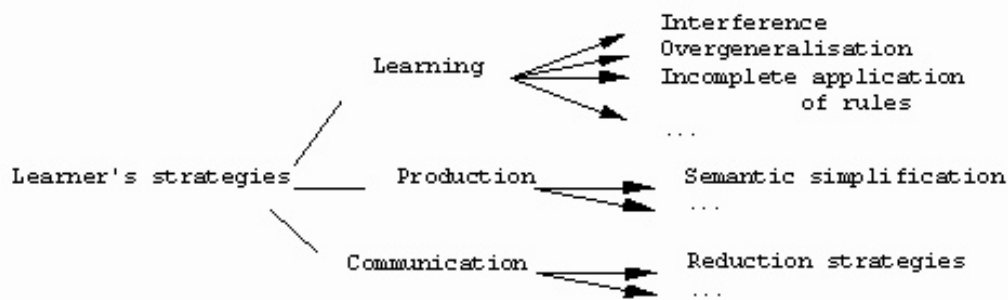

Figure 3.1: Simplified classification of the learner's strategies and the associated causes of errors.

Learner's Strategies of Learning

When the learner tries to create new knowledge about the target language, she first tries to make some hypothesis by using prior knowledge (of L1, L2 or another foreign language) or by inferring new rules from the input data on which she has to work. When relying on prior knowledge, she mainly uses two strategies: transfer of rules from the first language and generalisation (and overgeneralisation) of second language rules. Transfer and generalisation can be seen as a form of simplification. Simplification strategies consist of attempts by the learner to ease the burden of learning, to restrict hypothesis formation to those hypotheses which are relatively easy to form and will facilitate the communication. Once the learner has developed a hypothesis, she can test it out in a variety of ways such as: receptively (the learner attends to inputs expressed in the second language and compares her hypotheses with the data provided), productively (the learner produces L2 utterances containing rules representing the hypotheses she has formed and assesses their correctness in terms of the feedback received), meta-lingually (the learner consults a native speaker, teacher, grammar, or dictionary to establish the validity of the feedback received), or interactionally (the learner elicits a repair from her interlocutor). After having tested some hypotheses, part of the learning process involves consolidating them by accumulating confirmatory evidence, i.e. to practise L2 language either on formal aspects or on functional aspects. This is the automatisation process.

Some strategies used by the learner in order to simplify the task of learning L2 may have the side-effect of generating errors. Here is a description from [33]:

Language Transfer or Interference

---

[4] Interlanguage is the term coined by Selinker [37] to refer to the systematic knowledge of a second language which is independent of both the learner's first language and the foreign language.

The learner uses her previous mother-tongue (or possibly some other foreign language distinct from the target language) experience as a means of organising the second language. Here is an example of a transfer error from an English speaker in French:

> Je suis fait mon devoir *(Attempt to form a present continuous tense based on the pattern in English)*

It should be noted than the cause of this error is one with which every teacher is familiar, but is by no means the only one or even the most important (Contrastive Analysis asserted that this cause was the most important if not the only one, but studies done since the 1970s have indicated the contrary).

Overgeneralisation of the target language rules

The learner has acquired a particular piece of linguistic knowledge and some strategies which she had found helpful in organising the facts about the target language, but applies them too widely. This is a phenomenon frequently encountered.

> Who did write this book ? *(If this kind of sentence is uttered without any emphatic purpose, this means we have an overgeneralisation of the rule for inserting "do" into interrogatives.)*

Ignorance of rule restrictions

The previous example can also be explained as an ignorance of the restriction of the usage of "do" into interrogatives.

Incomplete application of rules

Incomplete application of rules involves a failure to learn more complex types of structure because the learner finds she can achieve effective communication by using relatively simple rules.

> Teacher: Ask her how long it takes.
> Student: How long it takes ?

False concepts hypothesized

False concepts hypothesized may derive from faulty comprehension of a distinction in the target language.

> he is speaks French *("is" may be understood to be the marker of present tense, as "was" is a marker of the past tense).*

Transfert on training

Errors may also come from the influence of a teacher and/or a textbook writer who, inevitably, emphasises some aspect of the target language and neglects others, according to his own beliefs and experiences. For example, teachers and materials place special emphasis on the present continuous form in English because they know that the learner generally does not have an equivalent form in her native language and may thus be inclined to ignore it. This presentation may have the undesired effect of making the learner overuse this form as in:

```
            In French we are not having a present continuous tense and we
            are not knowing when to use it.
```

Learner's Strategies of Production

Strategies of production relates to the unproblematic use of L2 knowledge, i.e. the learner relies on acquired knowledge to communicate. Starting from her communicative goal she will use planning strategies to create new utterances. Here again, she tries to simplify the task and thus errors may be generated, even if she has a correct knowledge. Two basic planning strategies can be identified: semantic simplification and linguistic simplification. For example, the learner can simplify a sentence by reducing the number of semantic cases and presupposing that the listener will infer the gaps, as in:

```
            Hitting me (Action + Patient)
            instead of :
            He is hitting me (Agent + Action + Patient).
```

Learner's Strategies of Communication

Strategies of communication occur when the learner has a gap in her knowledge and has to fill it or by-pass it in order to achieve the communication. In the second case, she can use *reduction strategies*, i.e. gives up part of his original communicative goal:

```
            He plays ... (the learner cannot find the name of the sport,
            thus does away with the problem and reformulate her sentence )
            He does sport.
```

In the first case, when she wants to compensate for insufficient means, she uses *achievement strategies* , such as the replacement of an L2 item with an item made up from L2 forms, e.g. utters "picture place" instead of "gallery".

3.3. Application of Error Analysis in ITS for SLL

The recommended incorporation of an Error Analysis approach into the teaching process is important for two reasons: firstly, it provides insights into the process of language acquisition; and, secondly, it de-emphasises the idea of errors as failure. Its disadvantage is that it is time consuming and suitable only for one-on-one tutoring and thus is an obvious candidate for computer-based teaching. The Error Analysis approach is also beneficial in determining remediation strategies since it identifies errors and gaps in the acquisition process itself as well as the acquired knowledge. However, this knowledge is not described in a formalized way that we could translate directly into rules. We need systems which could collect the expertise of expert teachers for that purpose (see section 5.4).

4. Overview of learner modelling

The aim of learner modelling is to provide a more responsive and individualised form of tutoring. Most of the modelling has been related to domain knowledge at the level of the observed performance. Generally speaking, a multi-level approach, incorporating information for the generation of the rules that directly produce the behaviour, has seldom been used. The use of information about a learner's meta-level rules we believe to be necessary for a cognitive diagnosis that can lead to more effective remediation and instructional methods. A general framework for the describing cognitive diagnosis with this level of detail can be found

in Dillenbourg and Self [15]. Here, we consider previous attempts to implement learner models and highlight some of the problems that have been encountered.

Modelling techniques

The two most common techniques for representing a learner in an ITS are *overlay* and *bug* modelling. The overlay model is applicable where the expertise can be expressed as a set of rules. The learner's state of knowledge is viewed as a subset of this expert rule set. Each rule is evaluated for the likelihood of the learner possessing it. A number of sources of evidence are used to assign a probability to the learner's possession of a rule; such as implicit sources by comparing the learner's behaviour with an expert's, or explicit sources by directly querying the learner about her beliefs. Even though overlay models have been a common structure for learner models they do have significant limitations; they assume that all errors are gaps in the learner's knowledge, as compared with the expert's, whilst it is clear that errors are produced by the use of distorted and replacement correct rules.The "buggy" approach to learner modelling tackles this problem by substituting variants of the correct rule or rules into the problem solving procedures. Generally speaking, in both these techniques the learner is represented as a set of production rules that, if applied to a problem, duplicate the learner's performance or behaviour. In the buggy models the rules that account for the learner's non-standard performance are usually drawn from a set of known possible rules, usually called a bug catalogue. This approach has been used in subject areas such as subtraction [6], fractions [29], and Lisp programming [2]. The development of the bug catalogues - which can run into several hundred rules - has been very labour-intensive. The implication is that it requires a great deal of time and expertise (through experimental studies and laborious analyses of learners' problem solving) to develop a bug catalogue which is adequate to support intelligent tutoring. Additionally, bug catalogues are concerned with relatively superficial differences in the behavioural models and not directly with the underlying misconceptions which gave rise to the bugs. The use of bug catalogues model the learner's problem solving as a set of uninterpreted syntactic rules; whereas it is likely that the syntactic manifestations are produced by the learner reasoning about the semantics of the problem. This lack of deeper representations limits the ITS 's ability to remediate successfully.

Generation techniques

The generation of bugs has been explained in a number of ways, for instance by Repair Theory [7] and by extrapolation techniques [25]. Repair Theory is intended to be domain independent, with new rules being produced by skipping steps and replacing steps by analogous ones in existing rules. Unrestricted use of repairs leads to some very peculiar errors that have not been observed in real learners. But again these operations are carried out at a syntactic level and an ITS is still unable to capitalise on the fact that the learner has actually attempted a repair at all. Laurillard [23] also points out that the method still fails to capture the necessary nature of the learner's misconceptions that can be used in selecting an appropriate teaching action. Matz [25] also has proposed a mechanism for the generation of bugs. The cause of errors in her theory (which has never been implemented) is the application of general extrapolation techniques to example problems in one domain from domains with which the learner is already familiar, and which under different circumstances would lead to correct actions. Matz cites the case of novices' attempts to transfer techniques from arithmetic to algebra. This requires knowledge of comparable domains to be encoded in the system with the concomitant effort. Although both these techniques would be applicable to mal-rule generation for SLL neither on their own has the power to produce the variety of rules that are

known to exist in SLL; although the extrapolation technique does appear to be the analogue of 'language transfer', described in the previous section. Hence it will be necessary to augment these techniques by more varied generation techniques and by the acquisition of expert rules that are a direct result of teaching experience.

5. Treatment of errors in SLL ITSs

5.1. Different approaches in SLL ITSs

The primary components of the construction of language competence and skills in communication are, among others, formal linguistic competence (knowledge of morphology, syntax, etc.) and functional competence (the ability to use the language to express meaning). Because of the difficulties encountered in natural language processing, currently functional and formal competences cannot be simultaneously cultivated with the same degree of accuracy in ITSs for second language learning. This implies, at least, two different approaches to modelling which are not related in the same way to the Error Analysis problem.

Functional approach

Since the focus is to facilitate the expression of meaning, ITSs based on the functional aspect of natural language [27,20,28] are not directly concerned with the formal correction of the learner's production. When a sentence cannot be parsed as correct, computational linguistics techniques may still be used but only to handle the ill-formed input and to manage the dialogue in order to try to understand the meaning of the utterance. The strategies of communication described in applied linguistics may supplement part of the learner model and make a link between this model and the computational linguistics techniques. Within these problem areas user modelling techniques used in computational linguistics [22] and learner modelling techniques in ITS seem to converge. This is an area of research which has not been much explored, since most of the systems dealing with the functional approach have, at the present stage, no learner model. But we will not pursue this learner modelling problem here since we are focusing on the formal approach.

Formal approach

With regards to SLL ITSs concerned with formal aspects of natural language, we will distinguish, for the sake of this presentation, two types of systems: *computational-errors systems* and *early deeper Error Analysis systems*. Broadly speaking, until recently, two competing factors seem to differentiate existing systems: the linguistic coverage versus the depth of the Error Analysis. Computational-Errors systems have, relatively speaking, a large linguistic coverage but depth limited Error Analysis procedures. We are mainly referring here to the variety of errors handled and, especially, the limited diagnosis of their causes. Early systems with a deeper Error Analysis had, on one hand, a narrow linguistic coverage but, on the other hand, a quite exhaustive description of the possible errors which may have occurred in their domains.

We present these two kinds of approaches, with their advantages and drawbacks, trying to differentiate the computational linguistics problems from the tutoring ones[5]. After this, ongoing research on capturing and modelling learner's processes will be introduced. This aims

---

[5] A different presentation of ITSs on SLL can be found in [40].

at filling the gaps highlighted in the two previous approaches and at moving towards a better error diagnosis.

5.2. Computational-Errors systems

A Computational-Errors system consists mainly of a computational grammar and a *computational-errors-parser*. A computational-errors parser is a parser which uses some computational strategies to recognise an utterance as erroneous, to locate this error with respect to the computational grammar and is, sometimes, able to help to remediate. The underlying teaching strategy in these systems is to show the learner her errors (with respect to the computational grammar) and possibly initiate a dialogue with her to help correct this particular utterance. Below we present some Computational Errors systems in terms of their skill in the computational diagnosis process.

The French Grammar Analyser

The French Grammar Analyser (FGA) [3] is one of the early system of grammatical error analysis. Here, every error has to be predicted and described explicitly by hand in the grammar. For example, if one is concerned with agreement problems between two categories, one has to foresee all the different situations in which such a problem may occur. The system cannot access the existing linguistic knowledge to diagnose problems. Some strategic choices appear to have been made without much concern for linguistic or pedagogic relevance (cf. procedures about unknown words). This may explain why the system proposes some wrong corrections and eventually accepts incorrect sentences.

The XTRA-TE system

The XTRA-TE system [12] is an ITS built on the top of XTRA, a translation Chinese-English grammar, which has a large coverage: 7000 English words, 10000 Chinese words, 500 grammar rules. Error diagnosis is handled by a multi-pass parser. Syntactic and semantic knowledge are separated in the grammar so that it is possible to distinctly relax some of the corresponding constraints. Instead of adding a new grammar rule for each possible error, definitions of grammatical categories are expanded in a traditional way (with respect to computational linguistics techniques). Figure 5.1 presents a simplified example of the definition of a verb phrase category which allows the identification of a non-agreement in number between the subject and the verb. It is given in the Definite Clause Grammar format.

```
sentence(Np, Vp) -->
noun_phrase(Np,Num),
verb_phrase(VP,Num).

verb_phrase(vp(v(V),np(Np)), Num) -->
trans_verb(V,Num,Relax_flag),
noun_phrase(Np).

trans_verb(V,Num, _) -->
t_verb(V,Num). (i)
trans_verb(V,Num, Relax_flag) --> (ii)
Relax_flag == true,
t_verb(V, WrongNum),
WrongNum \= Num,
error_flaggin(X,subject_verb_agreement,[Num, WrongNum]).

t_verb(likes,single).

In a correct sentence the verb is analysed by the clause (i). If there is a problem
of number agreement between the subject and the verb ("The old men likes golf"),
clause (i) will fail. After the detection of this failure the program sets the error
flag, Relax-flag to true and calls for another parse. The clause (ii) notes a
difference of number and stores the corresponding error.
```

Figure 5.1 : Relaxation constraints in XTRA-TE

The authors argue that this strategy avoids duplication of grammar rules, but some duplication still exists in the definition of grammatical categories. In this example all the definitions of verb phrases in the complete grammar have to be extended in order to tackle all the different cases of subject-verb disagreement. The extension of the whole grammar may be limited because they only have a full cover of subject-verb agreement and of pronominal errors. XTRA-TE is able to diagnose others types of errors but leaves aside the problem of incorrect word order which may imply an expansion of the grammar.

A learner model is built on the evaluation of the learner's errors which reflect her lack of certain concepts. A score is attributed to each learner. XTRA-TE uses this score to differentiate its strategies of correction during the session (indirect correction, direct correction, etc.).

Menzel's approach

Menzel presents an error diagnosis procedure [26] which tries to concentrate on an exact localization of rule violations and is able to infer information about factual faults as well. His approach is based on the following basic principles: (1) supply the system with only knowledge about correctness; (2) devise a diagnosis procedure which is independent of the content of the knowledge base. In order to achieve his aim he chose the restricted domain of the morpho-syntactic agreement in German.

When an error is encountered, his idea is to select a set of relevant constraints C, build a resolution proof on them which finds the minimum set of constraints which have been violated, and negate them in order to let C accept the divergent sentence. From this subset of negated constraints the system is then able to explain the failure in the parse and to propose a remediation.

This strategy avoids the duplication of rules in the grammar due to a change in the unification algorithm. It is an interesting approach but one which needs much more work in order to be incorporated in a large grammar. As a matter of fact, when the parser has to manage a great

number of rules, the selection of the relevant subset of constraints without any hint seems to be an intractable problem.

The ILTS for German system

The ILTS for German [36] handles a variety of errors and corresponding strategies in a way close to the XTRA-TE system, plus a treatment of certain word-order problems. A menu-based interface offers the learner the possibility of accessing part of the system's knowledge and of self-correcting some errors. The agreement errors procedure is quite original. Let us present it through the analysis of the divergent sentence:

```
Der Götter zümen (the gods are angry).
```

Within the unification grammar formalism of ILTS each word may be defined as a set of couples (feature, value). Hence the definitions of *der* and *Götter*:

```
der: {[Art-cat(def),Gender(masc),Case(nominative),Number(singular)],
[Art-cat(def),Gender(fem),Case(or(genitive,dative)),Number(singular)],
[Art-cat(def),Gender(_),Case(genitive),Number(plural)]}.

Götter: {[Gender(masc), Case(neg(dative)), Number(plural)]}.

der Götter: {[Art-
cat(def),Gender(masc),Case(genitive),Number(plural)]}.
```

There are two possible diagnoses of the error: (1) *Der Götter* is genitive plural (unification of *der* and *Götter*), but since it is the subject of the sentence, the right case should have been nominative. This is a disagreement on the case. (2) *der* is singular and *Götter* is plural ; this a disagreement on the number (the system can find this error since the unification process keeps the trace of all the pairs of value sets which have not been unified).

Schwind argues that the first case is improbable since it is harder, for the learner, to build a genitive case, which is a "difficult case", when the nominative is required, which is the "easiest" case. She explains this choice by the fact that "people make errors in order to make their life easier". But this relevant argument is not declared explicitly in the system as a learner strategy. The choice of (2) is made by a procedure which filters cases and may have no such concern.

Error diagnosis in Computational-Errors systems

In the Computational-Errors systems the ITS paradigm is encompassed by the juxtaposition of a computational grammar and a computational-errors procedure. Such procedures which try to locate the error within the computational grammar and find a way to repair the failure that occurred during the parse have been presented. This is a difficult computational problem which has not been fully achieved.

However, we think that, even if those techniques are necessary in a SLL system, they should not represent the final aim of an ITS. The computational grammar is not the grammar that the learner uses to learn (or the teacher to teach) a language (see section 6). Thus the diagnosis of an error and the corresponding explanations given by a Computational-Errors system reflect the system/parser behaviour but account very badly for the learner's use of her own grammar rules. Since the processing of an incorrect sentence may make the parser fail, the learner's errors are still considered as a failure in the learner's process of learning, reflecting a gap in

her knowledge with respect to the "complete" knowledge of the system. Thus the learner is never considered as an agent who creates new knowledge and the acquisition of a second language is never understood as an elaborative process[6].

5.3. Early systems with a deeper Error Analysis

Some SLL systems tried to achieve a deeper Error Analysis of the learner. In restricted domains errors have been analysed and explained with respect to criteria used in applied linguistics which may better reflect the learner's own process of acquisition. Two representative systems are presented.

The VP2 system

VP2 [35], an ITS dedicated to teaching English to a native Spanish speaker, uses, like the previous systems, a computational grammar. But here the emphasis is put on the explanation of some learner's errors. Arguing that computer-assisted instruction systems should include a model of relevant aspects of users' prior knowledge to predict and prevent errors or detect and correct them easily, the author decided to focus on interference problems. More precisely, she chose to work on the acquisition by non-English speakers of English verbal constructions formed from a verb plus particle or verb plus prepositional phrase. In this domain there is a large variation between English and Spanish languages.

The system is provided with grammars of both languages. The Spanish grammar models the learner behaviour. From this modelling the system is able to give some explanations to the learner, both whether a sentence is correct or incorrect. In this example of a correct answer, VP2 provides the learner with some further explanation and draws her attention to some underlying problems.

```
Tutor: Translate the following sentence:
yo soñé con los 'angeles.
Student: I dreamed of the angels.
Tutor: Correct! Note that the direct translation of <soñar
con>-<dream with>
does not exist in English. In English you can also use <dream
about>
in this sentence.
```

When a divergent sentence makes the parser fail, the system, after having used the English grammar, tries to parse it with the Spanish grammar. It then compares the two parse trees (English parse of the correct sentence, Spanish parse of the learner answer), identifies the wrong prepositions or missing particles and accesses the Spanish dictionary to explain the differences. An open question with this approach is whether it is possible to extend it to take into account other linguistic phenomena where there are deep differences between the source and the target languages. Another problem is how this treatment of interferences could be integrated within a more general model of the learner's interlanguage, the transfer strategies being only a subset of the learner's ones.

---

[6] Improvement of parsing techniques does not necessarily entail equivalent enhancement of the diagnosis of the learner's processes. eL [46], for example, is a much cleverer system than its predecessor, FGA: a declarative computational grammar, new parsing strategies enable the system propose corrections relating to word-order, agreements,... problems. But the learner's linguistic knowledge and its strategies have not yet been taken into account.

The ALICE system

ALICE [8] is a system which concentrates on the translation of temporal and causal conjunctions from Italian to English or French. This restricted domain has been chosen to investigate how a deep linguistic description could be associated with the understanding of conceptual difficulties by the learner.

The knowledge about the conjunctions is described quite exhaustively in a network of concepts. All the possible errors are tackled as well. They are divided in two categories: the ones which can be easily identified and linked with a predefined message of explanation (i.e. literal translation, confusion with adverbs, etc.), and those which need a deeper diagnosis process which will be achieved through a dialogue with the learner (i.e. interference, errors due to similarity, errors involving discrimination between concepts). In ITS terms, the diagnosis and correction of the errors in the first category rely on the use of a simple bug catalogue; those corresponding to the second category rely on a pattern matching process to find the closest learner model in a predefined set.

The answers given by the system to the learner, the characterisation of her errors are very precise and accurate at a linguistic level, but from a pedagogical aspect the learner's role seems to be passive during the whole session. In effect, it seems that after the learner has chosen a set of conjunctions to work on, the system will follow its own predefined sequence of exercises and will never take into account the learner's answers, or the set of learner's models which it has diagnosed as being close to the learner behaviour. The learner modelling does not seem to be reflected in the system's behaviour. Despite all the learner's answers being stored in the system for the teacher, the system cannot exploit them.

If the aim of ALICE were the validation, through a sequence of tests with learners, of the linguistic description of the chosen domain, this aim is accurately achieved. From an ITS standpoint, the current learner is not modelled specifically. The extension of the tools defined to associate a particular learner with a set of models to another domain may be problematic. Also, the extension of the linguistic knowledge may be difficult to achieve since no computational linguistics techniques are used.

5.4. Capturing and modelling the learner's processes

The systems mentioned above have addressed subspaces of the error diagnosis problem, but they lack generality and are thus hardly extendable to the whole space. Obviously, both a better understanding and a better formalisation of the learner's processes are required to extend the diagnosis space. During the last two or three years the research community in ITS for SLL has made inroads into these problems from a number of different perspectives.

Firstly, systems are being developed to support the acquisition of fundamental knowledge for the learner model from the various agents involved in language apprenticeship. The NOBILE system [9] allows experts, i.e. teachers, to design learner models through different kinds of interactions with the system, such as the "inverted dialogue" where the computer plays the role of the learner and produces divergent sentences and the user/teacher acts as the expert. The prototype IFAAR [30] offers an interface for knowledge acquisition from experts about learners' divergent sentences in order to capture the experts' diagnoses and the procedures they use. From the experiments undertaken with both experienced and trainee teachers, it transpires that these "experts" were not used to carrying out such deep and individual

diagnoses, and that current knowledge acquisition techniques used in expert systems were hardly applicable [43]. There is an urgent need in the ITS field for appropriate techniques to capture experts' diagnoses.

Secondly, prototypes that aim at effectively computing a deep diagnosis of the learner's processes are under development. The ET system [19] proposes a methodology to integrate bug catalogue and bug generation techniques, partly based on the use of explanation-based learning techniques, in order to get a deep modelling of the learner's performances in the apprenticeship of tense aspects of English. In the BELLOC system we propose a structure to represent the learner's rules and an architecture to manage a conceptual modelling of a learner's strategies (see next section).

Lastly, another perspective is the explanatory approach adopted in Zock's system [47] which allows the learner to explicitly build hypotheses about the interlanguage, to test them, and subsequently update her own rules. Even if no learner modelling is explicitly at stake in this system it provides insights on how language structures are learned. Interestingly, within this perspective, errors are not only considered as unavoidable parts of the learning process, but also as an indispensable source of information.

6. The BELLOC system

We now present the architecture of the part of BELLOC for building and updating the learner model through the conceptual diagnosis of her errors. We introduce the different components necessary to achieve this task and their inter-relations. The main component is the set of applicable rules which describes the teacher's as well as the learner's linguistic knowledge. Several kinds of ARs are necessary. We explain why and how each AR should be structured. Several examples indicate how they could be used in the diagnosis of the learner's behaviour.

6.1. General architecture

Figure 6.1 presents the general architecture which supports the handling of the conceptual diagnosis of the learner's rules with respect to the formal aspect of natural language. In order to introduce its main components we will sketch what happens from the time a sentence, which makes the parser fail, is entered until the diagnosis of the applicable rule which describes the problem and is relevant to the current user occurs.

When a sentence provokes a failure in the parser it means that some constraints in the computational grammar have been violated. A sequence of computational diagnostic procedures tries to give a partial diagnosis by identifying the relevant constraints. This corresponds to Part I in the Figure 6.1 of the architecture.
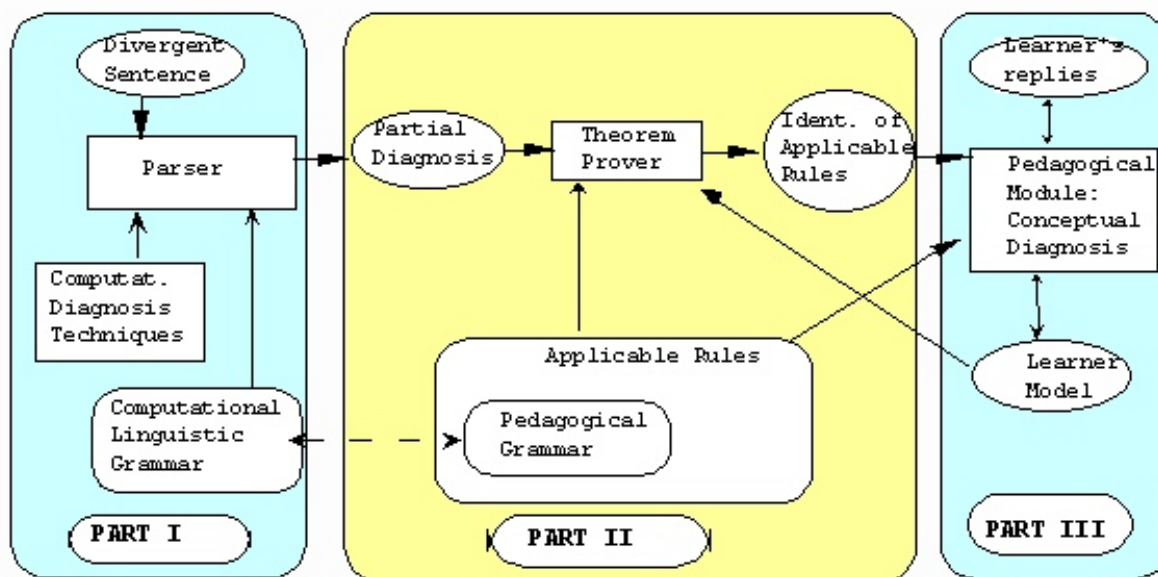
Figure 6.1 : Overview of the architecture for the treatment of errors in BELLOC. Ellipses represent dynamic information, Square rectangles procedures, and rounded rectangles static information. Dash arrows represent the relationships between static data.

Starting from this partial diagnosis a theorem prover identifies *candidate* applicable rules which could give a description of the learner's rule and strategies. The theorem prover searches among the set of applicable rules, some of them describing rules common to teachers and learners, others typical learners' rules, or general learners' meta-rules. During the deductive process, information coming from the previous state of the learner model can be used to prune the search space and make some preferences.

This process of identification of an applicable rule can be straightforward. In parallel with the call to computational diagnosis procedures, a second parse can be attempted with the computational linguistic grammar augmented by predefined learner's rules described in the same formalism. If one of these rules accepts the sentence, then an applicable rule is automatically found because of the one to one correspondence between these extra rules and a subset of the applicable rules. This corresponds to Part II in the Figure 6.1 of the architecture.

The candidate applicable rules are passed to the *pedagogical module* (see Part III of the picture of the architecture). It holds a dialogue (like the one in Figure 2.1) with the learner to check which ones apply effectively to him. This stage is what we call the *conceptual diagnosis*. During this process, information contained in the slots of the candidate ARs is used. Once it has diagnosed the AR(s) applicable to the current learner, the pedagogical module can update the *learner model* and decide to react (or not) in order to remediate or encourage, taking into account the aim of the working session (stress on communication, or formal aspects), the degree of seriousness of learner's divergence, previously related problems, teaching principles (see [42] on that point), etc. As a first approximation, the kinds of information relating to formal aspects of the target language that the pedagogical module would need to find in the learner model include at least: a set of ARs describing her linguistic knowledge and her strategies, possible fossilizations on some ARs, her level of knowledge of the meta-language necessary to talk about grammatical problems, history of previous identified ARs and of the dialogue. From now on, when detailing Part III of the architecture, we will restrict ourselves to the problem of diagnosing the relevant applicable rules, given an

empty learner model. The history of the learner and the pedagogical reactions will be left aside.

Divergent sentences

We call it a "divergent" sentence and not an "incorrect" one, because the appreciation of its level of correctness may vary from one learner to another. The sentence is divergent with respect to a *standard grammar*[7]. When a computational linguist develops a grammar, he tends to give a *correct exhaustive description* of a subset of a language. We refer to that knowledge of the language as the "standard" grammar. This standard grammar represents a viewpoint on a correct subset of a language, but by no means the definitive one, since one may have different views on what is correct or not in a given situation or the grammar may have some gaps.

Computational linguistic grammar and diagnosis techniques

The role of the computational linguistic grammar is controversial. Obviously we need such a grammar to parse the learner's inputs, to give her answers about the family database. In the IFAAR prototype, we have used a unification-based formalism, LUGA [21], to develop the syntactical grammar of a subset of French to cover the interrogative sentences of the learners. This formalism was chosen since unification grammars are a well tried and frequently used method of language representation (mainly syntactic) in the computational linguistics community (see [5,1,10] for example). But then two pointed questions arise: is the computational grammar formalism well suited to represent the linguistic knowledge of the learner, and how can we have a relevant diagnosis on the failure of the parse ? Let us try first to tackle the second question, i.e. the computational diagnosis techniques.

As pointed out in section 5, the description in the grammar of all the possible violations of constraints is unrealistic. The approach consisting of *modifying the unification process* in order to deduce what are the relevant violated constraints is certainly a promising area of research. But one needs some mechanisms to reduce the search space and help the unification solver to focus on preferable paths. Indeed even with a grammar which has a limited coverage several thousands of failures in the unification process occur in a successful parse! The second kind of diagnosis techniques which we can apply are the ones used in computational linguistics for the treatment of *ill-formed inputs* [13,45]. As illustrated in the example of diagnosis in Figure 6.4, an ordinary grammar cannot handle the semantic simplifications of the learner, which are quite frequent. Semantically-based techniques need to be used after the first parse, even on problems relating to formal aspects of natural language.

If we quote Shieber [38], one of the criteria against which to judge linguistic formalisms is linguistic felicity, i.e. the degree to which descriptions of linguistic phenomena can be stated directly (or indirectly) as linguists would wish to state them. This may be interpreted in two very different ways by computational (or theoretical) linguists and applied linguists. The former tend to describe and explain as fully as possible the phenomenon of language. The grammars they produce are by their very nature descriptive. Grammars used in applied linguistics are prescriptive. One lesson learned from the experiments with the IFAAR prototype is that a grammar developed on the traditional computational linguistics criteria is unreadable by expert teachers, even if explanations are provided. Secondly, one needs to

---

[7] We leave aside here divergences coming from slips, spelling errors and keyboard typing errors. For the treatment of these problems see [17].

incorporate in the ITS knowledge about linguistic problems usually encountered by learners of a given source language during their apprenticeship in the target language. It is encoded in what we call *predefined rules* (which represent one kind of applicable rule) and is not usually part of the background knowledge needed to develop a "traditional" computational grammar.

One way to bypass these drawbacks could be to rewrite the computational grammar with other criteria in mind, using a more restricted and accessible meta-language. We want to experiment with this option. But we think a more probable solution would be to also set out a pedagogical grammar.

Pedagogical grammar

We call "pedagogical grammar" the grammar created for instructional purposes in an ITS. It is supposed to be close to the one used by teachers during instruction. Obviously there exists no unique form upon which all teachers would agree. The question of teaching grammar explicitly or implicitly is still very controversial and answers vary much from one country to another. Some school text books offer a good picture of what a pedagogical grammar is: the words in which the linguistic knowledge is explained are carefully chosen and pedagogically presented (see Bescherelle for a reference in French[8] [4]). The rules are explained in plain natural language and significant examples and warnings about potential difficulties are explained

Swartz [41] has shown the central role that the design of the pedagogical grammar in a CALL system occupies, and the help that a hypertext tool can provide to support the structuring of this conceptual design in order to develop foreign language instruction. Unfortunately, as mentioned in her paper, the question of implementing this pedagogical grammar has been left aside at this stage. In fact this is a critical point. When one tries to formalise the knowledge presented in text books its incompleteness and even inconsistencies in the terminology are striking. On that aspect, research undertaken in Nijmegen provides elements of the answer. They have developed a pedagogical grammar, IPG, which can parse sentences and is used for Dutch grammar instruction [31,32]. They have adopted a consistent terminology to develop the grammar. Meaningful tree structures can be displayed to the learner with a level of detail depending on her mastering of the meta-language. For the purpose of learning a foreign language in a communicative environment, as we made the choice for BELLOC, it could be of interest to know if the IPG formalism can support the semantic features necessary to process the dialogue with our family database.

For the present, we have started to formalise a pedagogical grammar extracted from text books in order to describe it as a set of production rules. We want to investigate the links between this grammar and the computational linguistic one[9]. As we will see in the next paragraphs the pedagogical grammar is a subset of the set of applicable rules on which the theorem prover could operate in order to diagnose the learner's rules. Hence all pedagogical

---

[8] We hope the reader will understand that we are not committed to the definitions used in that book. We give it as an example of how a given linguistic terminology may be introduced for instructional purposes. Very different proposals have been made in CALL for the specification of this terminology: they range from a fixed vocabulary defined by psycholinguists in accordance with a theory [32] to an adaptation of the words effectively used by learners [14].

[9] In ET [18] the authors come to the same conclusion but from a different starting point. The first version of ET partly relied on a pedagogical grammar directly extracted from the literature ("naive grammar"); the second version will benefit from the coupling of this naive grammar with the formal systemic one in order to obtain a tractable and neat pedagogical grammar.

grammar rules will have a common structure in which their computational code (the production rule) will be contained in the "computational code" slot.

6.2. What is an applicable rule ?

The concept of an applicable rule is based on a number of ideas. Firstly, we want to separate the grammatical explanation which characterises the divergence of a sentence from the explanation that identifies the strategies the learner used to generate it. Next, we wish to move away from the concept of a mal-rule, which tends to emphasise the idea of failure, and move towards a concept that recognises the generation and updating of the language production rules as signs of learning. The de-emphasis of absolute correctness is important for other reasons. It is important to recognise that teachers also use applicable rules as approximations to the desirable performance as pedagogic devices to simplify the problems of learning. A common frame to describe the learner's and the teacher's rules would give a fairer reflection of this situation. Lastly, the domain of language is less clear cut than parts of mathematics and physics, traditionally used in ITSs. In consequence absolute correctness does not have any meaning. For example, the sentence "Combien de frères est-ce qu'il a?" would be considered incorrect in written French but is acceptable as spoken French.

The definition for the structure of an applicable rule, as described here, arose from these ideas refined by experimentation with the IFAAR prototype. During the elaborative process we tried to find answers to the following questions: how can we represent better the learner's state of understanding in the domain? what sort of linguistic and pedagogical knowledge does an expert bring to bear on the problem of diagnosing a learner's performance? in what terms would the expert prefer to express himself? what formalisms and implementations are suitable for encapsulating this knowledge? and what information is needed to discriminate between competing applicable rules that explain a particular behaviour. This gave rise to a six-slot structure for an AR (Figure 6.2), that we now detail.

Applicability slot: an AR is either applicable to the learner (a learner's rule), or to the teacher (a teacher's rule) or to both at the same time. In the first case it characterises the learner's performance, and, in the second, what the performance should have been. The third case will be illustrated in the last example of section 6.3.

Explanation slot: in a learner's rule the explanation slot contains a causal explanation of the learner's behaviour and optionally an explanation of grammar rules that could be used in the production of a sentence. In a teacher's rule, this slot contains a pedagogical explanation of the grammar rule expressed in meta-linguistic terms selected from a set generally used by the teacher and understandable by the learner.
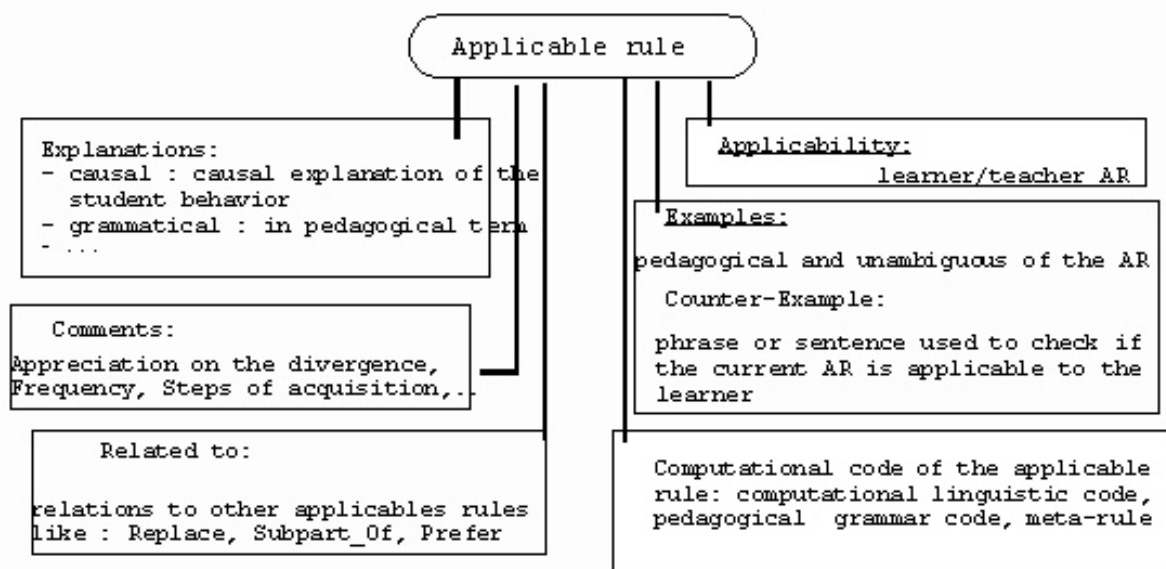
```
                          ╭─────────────────────╮
                          │   Applicable rule   │
                          ╰─────────────────────╯

┌──────────────────────────────────┐       ┌──────────────────────────────┐
│ Explanations:                     │       │ Applicability:               │
│ - causal : causal explanation of the│     │            learner/teacher AR│
│   student behavior                │       └──────────────────────────────┘
│ - grammatical : in pedagogical term│     ┌──────────────────────────────┐
│ - ...                             │       │ Examples:                    │
└──────────────────────────────────┘       │ pedagogical and unambiguous of the AR│
                                            │  Counter-Example:            │
┌──────────────────────────────────┐       │                              │
│   Comments:                       │       │ phrase or sentence used to check if│
│ Appreciation on the divergence,   │       │ the current AR is applicable to the│
│ Frequency, Steps of acquisition,..│       │ learner                      │
└──────────────────────────────────┘       └──────────────────────────────┘

┌──────────────────────────────────┐       ┌──────────────────────────────┐
│     Related to:                   │       │ Computational code of the applicable│
│                                   │       │ rule: computational linguistic code,│
│ relations to other applicables rules│     │ pedagogical  grammar code, meta-rule│
│ like : Replace, Subpart_Of, Prefer│       └──────────────────────────────┘
└──────────────────────────────────┘
```

Figure 6.2: Structure of an applicable rule. Each box explains the type of information placed in the slot. Slot types are underlined.

Example slot: because a number of applicable rules may be capable of explaining a particular behaviour, it will be necessary to disambiguate them if effective remediation is to be carried out. Typically, a teacher does this by the use of examples and counter examples in a dialogue with the learner. Thus the AR structure should allow for the inclusion of sentences and phrases that both correspond to the current rule and to other rules that produce similar but not necessarily identical behaviour. These examples may also be used to illustrate points of grammar in later remediation or instruction. For the present time we have a restricted use of the corresponding slot. Examples convey information to the learner after the conceptual diagnosis and to the developer of the computational grammars (linguistic and pedagogical). The examples should be pedagogically and unambiguously selected with respect to those presented in related ARs. The counter-example is a discriminatory one, it allows the test of the applicability of the current AR to the learner.

Related-to slot: although a general network of explanations may not be established there are a number of obvious links between explanations. The most obvious is the relationship between rules that explain the same learner behaviour. Another potential link is between rules that have some superficial resemblance, such as rules that require agreements between various parts of speech in the foreign language that are not required in the native language. There are also obvious links between rules that have specialisations and exceptions and naturally form themselves into a hierarchy. Therefore part of the structure of the applicable rule must represent the relationships with other applicable rules. The main kinds of relations, which will be illustrated in the following examples, are those linking teacher's rules and those linking learner's rules to teacher's ones.

Computational-code slot: how an applicable rule relates to a particular divergent sentence and how it can be automatically selected as a possible candidate to explain the divergence is described in the computational-code slot. Since ARs can be of very different sorts, procedures can be expressed in the computational linguistic formalism, or in the pedagogical grammar formalism or as meta-rules. The coupling of a divergent sentence to an applicable rule is either straightforward (the linguistic computational code successfully parses the sentence) or is the result of a deductive process.

<u>Comments slot:</u> lastly the structure of the applicable rule should be able to represent ad hoc information that teachers commonly have about certain errors, such as at what stage of learning they are mostly likely to occur, possible remediation techniques, and the seriousness of the error. This informal information fills the comments slot.

6.3. Examples of AR diagnosis

Through different examples we introduce three kinds of applicable rules and how they are used for conceptual diagnosis. Figure 6.3 presents the three applicable rules used in dialogue1.

AR1 is a learner's rule describing an interference/transfer problem (special case of learning strategy, see Figure 3.2). The learner tries to apply the English rule about possessor gender agreement to French sentences including possessive adjectives, as in the example: "Jean vit avec son soeur " (*Jean lives with his sister*[10]) . The computational code is written in LUGA. This code is included in the computational grammar, but can be fired only after a first failure in the parser and then will accept the divergent sentence. The use of the counter-example is twofold: firstly, the LUGA code should fail on this sentence; secondly, it serves as a discriminatory example during the dialogue with the learner. If she accepts "Cecile vit avec son frère", the system will conclude that she does not have this rule in mind. AR1 is what we call a *predefined rule*. This rule is part of the knowledge of teachers in foreign language. It has to be described explicitly in advance by hand. Any automation of their production is hard to foresee. It is the aim of the IFAAR prototype to capture them.

```
AR1, Learner's rule
Name: Possessor Gender Agreement
Examples:
Jean vit avec son soeur.
Mary vit avec sa frère.
Counter-example:
(Cecile lives with her brother.)
Cecile vit avec son frère.
Causal explanation:
Learning strategy, Transfer problem:
In English gender agreement with the
possessor. In French gender and number
agreement with the possessed object,
person and number agreement with the
possessor.
Computational code:
#define relations
Possessor_Gender_Agreement
< syncat> = [Possessor, Possessed]
< Possessor head agree gender> =
< Possessed head agree gender>
Related rule: Replace by: AR4, Possessive
Adjective Agreement.
```

```
AR2, Teacher's rule
Name: Possessive Adjective Morphological
Agreement
Examples:
ta soeur, ton autre soeur
ma belle héritière, mon héritière
Counter-example:
ta autre soeur
Pedagogical explanation:
the masculine forms "mon", "ton", "son"
are found qualifying a feminine
singular noun, when they are followed
by a word beginning with a vowel
or h-mute.
Computational code: production rule
Related rule: Subpart of : AR4,
Possessive Adjective Agreement.

AR3, Learner's rule
Name: Incomplete application
of rule <RULE>
Causal explanation:
Learning strategy, simplification
Computational code: ...
```

Figure 6.3: Applicable rules related to dialogue1 (see Figure 2.1) including a predefined rule (AR1), a meta-rule (AR3), and a pedagogical grammar rule (AR2).

AR2 is a teacher's rule and it is a *pedagogical grammar rule*. It describes the morphological agreement with possessive adjectives which is a special case of the possessive adjective agreement (rule AR4). Its computational code is a production rule because AR2 is a subpart of the pedagogical grammar. The direct relation with the LUGA code is not obvious because

---

[10] Standard French version should be:"Jean vit avec sa soeur " because "soeur" is feminine.

the agreement constraints may be expressed through several LUGA rules[11]; except that the standard LUGA grammar (i.e. the one which does not contain any predefined learner's rules) should accept the examples and reject the counter-example. "ta autre soeur" is also a discriminatory example. Its acceptance by the learner implies a rejection of AR2.

AR3 is a *meta-rule* describing a particular general learner strategy consisting of the incomplete application of rules in order to simplify the learning process. This rule is fired if a pedagogical rule ARn which is a special case of another one has been rejected. Then the variable <RULE> is instantiated with the name of ARn. This strategy may be a consequence of the teaching process. When introducing a grammatical problem the teacher can decide, as a first step, to leave aside special cases such as morphological agreement. This simplification can therefore induce the situation encountered here.

How does the system use these rules to arrive the diagnosis of figure 2.1? The analysis of the sentence "Que est sa adresse?" provokes a failure in the parser. A second parse is accomplished successfully with inclusion of predefined rules. AR1 accepts the phrase "sa adresse". The computational diagnosis procedures give partial information about the failure in the first parse to the theorem prover; from this information the theorem prover is able to select AR2 as the rule that has not been applied. It is also able to show that the application of the meta-rule, AR3, to the higher level rule AR4, of which AR2 is a subpart, produces the same effect. By the use of the counter examples in AR1 and AR2 the system deduces that AR3 is the appropriate learner's applicable rule and is able to offer the teacher's applicable rules AR2 and through the relationship slot the higher level rule AR4 that fully describes possessive adjective agreements.

Through this example we have presented the three types of applicable rules: predefined rules, pedagogical grammar rules, and meta-rules. Meta-rules, such as AR3, are very important. They allow the generation of specific learner's ARs from the pedagogical grammar. This approach may give the answer to one of the drawbacks of the mal-rule approach described in section 4, that of having to describe by hand each individual learner's rule.

```
Dialogue2

L: Qui sont les frères de Jean ?
S: Robert et Tom.
L: Qui est marié à Tom ?
S: Anne.
L: Est-ce que je le fils de Tom ? (i)
S: There is a problem in your last sentence.
Would you accept this utterance (yes/no) ? :
Est-ce qu'il est vivant ?.
L: yes
S: OK.I think the problem is the following:
Even if your utterance is comprehensible, there is a missing
constituent of <the verb etre>
```

Figure 6.4 : In the divergent sentence (i), the verb "to be" has been omitted.

A semantic simplification

---

[11] On such a simple example it is obvious that we could have a one to one correspondence between the computational rule and the pedagogical one. But if such a correspondence could be guaranteed at the level of the whole grammar there are few reasons to distinguish them. As we said, this is a point we want to investigate and so far, our experiments on more complex examples have found that strict correspondence was hard to maintain.

In dialogue2 the learner has omitted the verb "to be" (Figure 6.4, sentence (i)). The standard form should have been "Est-ce que je suis le fils de Tom ?". Two applicable rules are candidates (Figure 6.5). AR5, which is a predefined rule, states that the learner may have confused the interrogative phrase "est-ce que" with the verb "to be" (compare the word order in sentence (i) with the English one in the translation "Am I Tom's son"). AR5 is encoded as a LUGA rule.

```
AR5, Learner's rule                AR6, Learner's rule
Name: Omission of "etre" with "Est- Name: semantic simplification of
ce que"                            <CONSTITUENT>
Example:                           Causal explanation:
Est-ce que Richard le fils de      production strategy
Cécile ?                           omission in a sentence of
Counter-example: Est-ce qu'il est  <CONSTITUENT> =
vivant ?                           (verb, auxiliary, subject, DOC ,SOC)
Causal explanation: Learning       Pedagogical explanation:
strategy, false concept            Even if your utterance is
hypothesized                       comprehensible,
"Est-ce que" performs the function there is a missing constituent of
of the verb to be                  <CONSTITUENT>
Computational code:...             Computational code: ....
```

Figure 6.5 : Applicable rules relating to dialogue2.

AR6 is a meta-rule describing the semantic simplification of a constituent in a sentence like the verb, subject, direct object complement. The constituent(s) deleted are not syntactic but semantic. They consist of cases such as agent, object, patient, etc... The detection of the missing element is undertaken by special kinds of computational diagnosis procedures, fired after the first parse, which we referred to as treatments for ill-formed input.

Again the diagnosis between the two candidates rules begins with the most specific one, AR5. This is rejected after the acceptance of its counter-example. Since AR6 has no example/counter-example, nor is it related to any other rule, it is always an acceptable candidate. After the diagnosis the following rule will be added to the learner model:

```
AR6.1, Learner's rule
Divergent sentence: Est-ce que je le fils de Tom ?
Name: semantic simplification of <the verb être>
Causal explanation:
production strategy
omission in a sentence of <CONSTITUENT> = verb
...
```

A learner's/teacher's applicable rule

The diagnosis in Dialogue 3, shown in Figure 6.6, is accomplished by a rule that is both applicable to the learner and the teacher. Although sentence (i) is, on the one hand, a cumbersome construction, because it includes a double interrogative phrase, and is therefore not recommended in writing, it is, on the other hand, very frequently used by the French in speech because it avoids the difficult subject-verb inversion.

```
Dialogue3

L: Qui est mon père ?
S: Richard
L: Combien de frères est-ce qu'il a? (i)
S: There is a problem in your last sentence.
```

```
I think the problem is the following:
you can say : Pourquoi est-ce que Robert a hérité ? (ii)
but it is better to say : Pourquoi Robert a-t-il hérité ? (iii)
Explanation (yes/no) ?
L: yes
Good, you seem to know a construction frequently used in speech,
but be careful not to use it in formal writing,...
```

Figure 6.6 : The correctness of a sentence is a relative notion as illustrated in this dialogue.

Thus this double interrogative phrase construction is part of the the teacher's grammar, and therefore should be expressible within the pedagogical grammar, but it is also one of the learner's rules, in that it describes a particular type of simplification, and hence could be encoded as predefined rule in the computational linguistic grammar. This rule would be related to the pedagogical rule that calls for a "repeated subject in version in a constituent question" through a "preference" link. The necessity of updating the learner model with this rule would be a decision for the pedagogical module to take, probably based on past observations.

7. Conclusion and future work

One important challenge for CALL systems which tackle the formal aspects of second language learning is the modelling of the ways the learner handles heuristics and strategies to acquire and use the rules of the target language. Research in applied linguistics has shown that Error Analysis provides the basic elements on which to build relevant systems.

The recent developments in ITSs in second language learning have begun to demonstrate how it is possible to bypass the limitations of the traditional CALL systems and have pointed out how crucial the modelling of the learner is. However, the modelling of the learner's errors still remains superficial in systems which do not restrict themselves to a narrow linguistic corpus, i.e. systems which immerse the apprenticeship within a fruitful communicative environment. The extension of the linguistic coverage of a CALL system requires the use of computational linguistic techniques. The current state of the art in computational linguistics does not offer satisfactory answers to the modelling of the learner through the treatment of her errors. It is hard to develop a meta-level on top of the computational linguistic grammar which can at the same time handle divergent sentences and be relevant to the learner's processes.

In order to conceptually model the learner's errors it is indispensable to find out what is the right level of description of her rules and what are her strategies. Through the presentation of the architecture of the BELLOC system we have discussed a set of requirements which can help to fulfil these aims. We emphasised the need of a pedagogical grammar, beside the computational linguistic one, which plays a central role in the diagnosis.

We introduced the notion of an applicable rule, a structure which aims at representing learner's rules, as well as teacher's ones. This degree of equivalence between the learner and the teacher de-emphasises the learner's failures and draws closer links between the learner's process of production and refinement of her rules, and the teacher's process of simplification of his knowledge for pedagogical purposes. By examples, we illustrated how an ITS can lead to a deeper conceptual diagnosis of the learner's behaviour. The information that applicable rules encapsulate outlines the strategies a learner uses and thus provides the learner model with a fundamental source of knowledge.

In applicable rules the implementation level is clearly separated from the conceptual one. Despite the very different ways they can be implemented, and the different strategies they use for recognising a divergent sentence, they all have the same general structure. We distinguished three kinds of applicable rules. Firstly, predefined rules recognise divergent sentences relating to specific linguistic difficulties that only an expert teacher can predict. When considering the computational code of a predefined rule, expressed in a computational linguistic format, one can compare it with the mal-rule approach used in other ITSs on SLL. Secondly, pedagogical grammar rules correspond to the teacher's rules. Being part of the objective of the apprenticeship, they are expressed in a formalism close to the one used by teachers during instruction. Thirdly, the meta-rules describe general learner's strategies. They allow automatic generation of particular applicable rules specific to the current learner in close relation (for part of them) with the pedagogical grammar.

We have presented here what we believe is a novel approach to the problem of cognitive diagnosis in ITSs for SLL. It is based on established techniques in applied linguistics and on observations of expert teachers analysing the performance of learners. We should emphasise that the framework presented is a conceptual model only. The detail, we hope, is sufficient to provoke discussion on both representational issues of learner modelling and on architectural aspects of encapsulating the diagnosis process.

In the short term, we intend to implement a pedagogical grammar and use it to build learners' applicable rules from the partial diagnoses coming from the computational linguistic part of the BELLOC system. Also, experiments with learners will be set up in order to validate and refine the user modelling process by building applicable rules within a dialogue game interaction.

Acknowledgements

Bibliography

1. Abeillé A.(this volume): "A lexicalized Tree Adjoining Grammar for French and its relevance to language learning".

2. Anderson J.R. and Reiser B. (1985): "The Lisp tutor". *Byte*, vol. 10, pp. 159-175.

3. Barchan J., Woodmansee B.J., and Yazdani M.(1986): "A PROLOG-based Tool for French Grammar Analysis". *Instructional Science*, vol. 14, pp. 21-48.

4. Bescherelle (1984): *Le nouveau Bescherelle, 3. La grammaire pour tous*. Paris: Hatier.

5. Boguraev B., Carrol J., Briscoe T., and Grover C. (1988): "Software Support for Practical Grammar Development". *COLING 88* , Budapest.

6. Brown J.S. and Burton R.R. (1978): "Diagnostic models for procedural bugs in basic mathematical skills". *Cognitive Science*, vol. 2, pp. 155-192.

7. Brown J. S. and van Lehn K. (1980): "Repair theory: a generative theory of bugs in procedural skills". *Cognitive Science*, vol. 4, pp. 379-462.

8. Cerri S.A. (1989): "ALICE: Acquisition of Linguistic Items in the Context of Examples". *Instructional Science* , vol. 18, pp. 63-92.

9. Cerri S., Cheli E., and McIntyre A.(this volume): "Nobile: Object-Based User Model Acquisition for Second Language Learning".

10. Chanier T. (1989): *Compréhension de textes dans un domaine technique : le système ACTES. Application des Grammaires d'Unification et de la Théorie du Discours*. PhD Dissertation, Université Paris-Nord, Villetaneuse, June.

11. Chanier T., Dillenbourg P., Hartley R., Hinzte D., Nichols D., Pengelly M., Self J., and Twidale M. (1990): *A Prototype for the Acquisition of Rule-Based Student Models*. AI-Report n° 40, Computing Department, University of Lancaster, January.

12. Chen Li and Barry L.K. (1989): "XTRA-TE: Using Natural Language Processing Software to Develop an ITS for Language Learning". *4th International Conference on Artificial Intelligence and Education*; Amsterdam, May, pp. 54-63.

13. Computational Linguistics (1983): Special Issue on Ill-Formed Input. *American Journal of Computational Linguistics*, vol 9, 3-4.

14. Demaiziere F., Blanvillain O.(this volume): "Learners' intuitions, misconceptions and errors".

15. Dillenbourg P. and Self J. (1990): *A Framework for Cognitive Diagnosis*. AI-Report n° 48, Computing Department, University of Lancaster.

16. Ellis R. (1985): *Understanding second language acquisition.*. Oxford: Oxford University Press.

17. Fouqueré, C. (1988). *Systèmes d'analyse tolérante du langage naturel* . PhD Dissertation, Université Paris-Nord, Villetaneuse, January.

18. Fum D and Tasso C.(this volume): "Naive vs. Formal Grammars: A case for integration in the design of a foreign language tutor".

19. Fum D., Giangrandi P., and Tasso C.(this volume): "The use of explanation-based learning for modelling student behavior in foreign language tutoring".

20. Hamburger H. and Hashim R.(this volume): "Foreign Language Tutoring and Learning Environment".

21. Johnson R. and Rosner M. (1989): "A Rich Environment for Experimentation with Unification Grammars". *Proceedings of the European Association for Computational Linguistics*, Manchester.

22. Kobsa A. and Wahlster W. (eds.) (1989): *User Models in Dialog Systems*. Springer-Verlag.

23. Laurillard D. (1988): The pedagogical limitations of generative student models. *Instructional Science* 17, 29-55.

24. Littlewood, W.T. (1984). *Foreign and Second Language Learning. Language acquisition research and its implications for the classroom*. Cambridge University Press.

25. Matz M. (1982): "Towards a process model for high school algebra errors". In D.H. Sleeman and J.S. Brown (eds.), *Intelligent Tutoring Systems*, New York: Academic Press.

26. Menzel W. (1988): "Error Diagnosis and Selection in a Training System for Second Language Learning". *COLING 88*, Budapest, August.

27. Mulford G.W. (1989): "Semantic Processing for Communicative Exercises in Foreign-Language Learning". *Computers and the Humanities, Special Issues on the Intelligent Computer-Assisted Language Instruction*. Vol. 23, 1, January.

28. Murray J., Morgensten D., and Furstenberg G. (1988): "The Athena Language Learning Project: Design Issues for the Next Generation of Computer-Based Language Learning Tools". In Smith W.S. (ed.), *Modern Technology in Foreign Language Education: Applications and Projects*. American Council on the Teaching of Foreign Languages.

29. Nwana H.S. and Coxhead P. (1988): "Fraction bugs: explanations, theories and implications for intelligent tutoring systems". *Proc. of 6th Workshop of European Soc. for the Study of Cognitive Systems*, Varena.

30. Pengelly M., Chanier T. (1990): "Knowledge Acquisition for learner modelling in second language learning". Paper presented at the *NATO advanced educational technology workshop 'Cognitive Modelling & Interactive Environments'*. Eindhoven, The Netherlands.

31. Pijls F., Daelemans W., and Kempen G. (1987): "Artificial intelligence tools for grammar and spelling instruction". *Instructional Science*. vol 16, pp 319-336.

32. Pijls F., Kempen G., and Janner E. (1989): *Intelligent modules for Dutch grammar instruction*. Report of the Experimental Psychology Department. University of Nijmegen. The Netherlands.

33. Richards J.C. (1974). "A non-Contrastive Approach to Error Analysis". In [34].

34. Richards J.C. (1974): *Error Analysis: Perspectives on Second Language Acquisition*. Longman. Reprinted in 1984.

35. Schuster E. (1986): "The role of native grammars in correcting errors in second language learning". *Computational Intelligence*, vol. 2, pp. 93-98.

36. Schwind, C. (1990): "An Intelligent Language Tutoring System". To appear in *International Journal of Man-Machine Studies*.

37. Selinker L. (1974): "Interlanguage". In [34].

38. Shieber S.M. (1986): *An introduction to unification-based approaches to grammar*. CSLI Lecture Notes. University of Chicago Press.

39. Swan M.and Smith B. (1987): *Learner English. A teacher's guide to interference and other problems*. Cambridge University Press, UK.

40 Swan P (1990): *Theory and Practice of Computer-Assisted Language Learning*. PhD dissertation, Open University, Milton Keynes, UK, August.

41. Swartz M.L.and Russel D.M. (1989): "FL-IDE: hypertext for structuring a conceptual design for computer-assisted language learning". *Instructional Science*, vol. 18, pp. 5-26.

42. Swartz M.L.(this volume): "Issues for Tutoring Knowledge in Foreign Language Intelligent Tutoring Systems: Knowledge Representation and Communicative Language Teaching".

43. Twidale M.B. (1990): "Knowledge Acquisition for Intelligent Tutoring Systems". Paper presented at the *NATO advanced educational technology workshop 'Cognitive Modelling & Interactive Environments'*. Eindhoven, The Netherlands.

44. van Els T., Bongaerts T., Extra G., van Os C., and Janssen-van Dieten A.M. (1984): *Applied Linguistics and the Learning and Teaching of Foreign Languages*. London: Edward Arnold.

45. Wilks Y. and Farwell D.(this volume): "Description of the Intelligent Second Language Tutor Application".

46 Yazdani M., O'Brien P., Byron G.(1989): *An English Tutor: Project Report (1987-1989)*. Report of the Computing Department, University of Exeter, UK.

47. Zock, M.(this volume): "SWIM or sink : The problem of communicating thought. How can I tell you how to say it, if you can't tell me what you want to say?".