SPEECH SYNTHESIS FOR LANGUAGE TUTORING SYSTEMS

Rodolfo Delmonte

> *"It would be a considerable invention indeed, that of a machine able to mimic speech, with its sounds and articulations. I think it is not impossible."*
>
> Leonhard Euler (1761)

I. INTRODUCTION

In this paper we shall be concerned with the use of Text-to-Speech Synthesis, or TTS for short, as a tool for Language Learning. We shall present a number of applications where TTS plays a fundamental role in helping the student in Second Language learning. TTS can be a fundamental tool in helping to recognize and get aware of the contrastive features that constitute the main learning targets of the student. TTS can also be used simply as a speaking Tutor when help is needed in any self-instructional system or just to provide feedback on some exercise the student is practicing. It can be used as a Reader for Dictation exercises where there is a need to vary voice quality and speaking rate. Eventually, it can be used to help students working on a Listening Comprehension task in giving hints on what the main task to be accomplished consists of, and other similar Oral drills.

We shall be presenting all these examples of the use of TTS in a CALL without always assuming that it is the only way to cope with oral linguistic practice. In general, having a human tutor to do the same kind of tutoring activity guarantees a much better result: the question is whether a human tutor may always be available when the student needs one, which is usually not the case. So the possibility to have a substitute, for how much of lesser quality it may be, is worth pursuing. And there is at least one case in which the computer-based speaking tutor constitutes the only viable alternative to the human tutor: when mimicking the levels of speaking proficiency in L2, or levels of interlanguage, as will be explained in detail further on.

II. TTS and ASR - Two technologies in comparison

TTS has been around for quite a number of years now, from about the end of 60's. However, only in the last four or five years it has become a companion to most computer's systems and applications. Thanks to its easiness of usage and implementation, tutoring systems are more and more approaching a human-like appearance because of the presence of TTS applied to virtual talking heads or agents.

This notwithstanding, TTS has not yet reached a level of technological maturity comparable to that of Automatic Speech Recognition (hence ASR) which attained its apex when announcing Continuous Speech Recognition with Large Vocabularies beginning of the 90's.

However, we feel that, as happened in ASR, a combination of phonetic and linguistic structural information with proper use of statistical procedure will eventually lead to improved results in TTS.

And it is a fact that a number of important achievements have been achieved, especially in these latter years:

a) There has been an increasing convergence on which approaches to use which has lead on the one hand to the extension of the linguistic coverage to an impressive number of different languages [see the quote below];

b) A number of websites offer a low-level standardized full-fledged TTS system for free with a number of accompanying tools to develop the modules needed to improve naturalness besides intelligibility, and one of those is the EULER Project where I took the quote by Euler from [1];

c) As D.O' Shaughnessy remarked in his review on *Computational Linguistics 24, 4*, [2] it has been more than a decade since a comprehensive couple of textbooks readers and collection of very advanced papers on TTS has been published: and this happened just in the last two years [3];

d) Dialogue tutoring systems have introduced Virtual Talking Heads equipped with such standardized TTS systems with an intelligible quality level - we are here referring especially to the use of Festival in the CSLU Toolkit [4].

2.1 The question of naturalness

Even though the introduction and practical usage of TTS in Dialogue and Tutoring Systems is an encouraging result, we feel that the main issue constituted by improving the quality of speech is thus disregarded. Pros and cons of this kind of attitude towards TTS may be summarized by using R.Sproat and J.van Santen opinions as expressed in the Introduction to [5]:

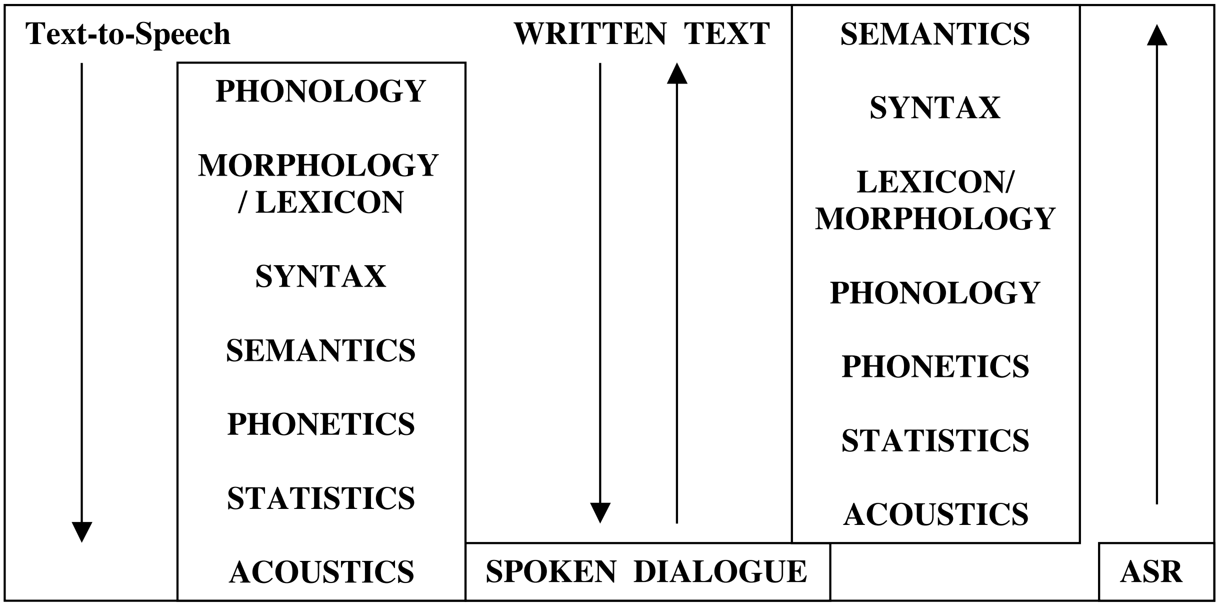
> "... We feel it is nevertheless important to point out that the ultimate goal - that of accurately mimicking a human speaker - is as elusive as ever and that the reason for this is no secret. After all, for a system to sound natural, the system has to have real-world knowledge, know rules and exceptions of the language; appropriately convey emotive states; and accurately reproduce the acoustic correlates of intricate motor processes and turbulence phenomena involving the vocal cord, jaws and tongue. What is know about these processes and phenomena is extremely incomplete, and much may remain beyond the grasp of science for many years to come.
>
> In view of this, the convergence in current work on TTS is perhaps somewhat disturbing. For example a large percentage of current system use concatenative synthesis rather than parametric/articulatory synthesis. We believe that this is not for theoretical reasons but for practical reasons: the quality levels that are the current norm are easier to attain with a concatenative system than with a parametric/articulatory system.
>
> However we feel that the complex forms of coarticulation found in human speech ultimately can only be mimicked by accurate articulatory models, because concatenative systems would require too

> many units to achieve these - significantly higher - levels of quality....
> We make these remarks to express our belief that TTS is not a mature technology. There may exist standard solutions to many TTS problems, but we view these solutions only as temporary, in need of constant scrutiny, and if necessary rejections." [ibid., 1-2]

We made this long quotation from [5] because we fully agree with the two authors' position and we feel that the question of improving the quality of TTS requires too long a discussion to be even vaguely introduced here.
We will only lay out the foundations of TTS as far as this can help the reader to better understand the subsequent part of this article.

2.2 How is TTS achieved

We made a comparison at the beginning between TTS and ASR: the two technologies can be regarded - theoretically at least - as integrating one another's internal architecture. It is a fact that in order to generate an appropriate message for the user to serve as feedback in a given tutoring situation, a Dialogue System that uses TTS needs a certain number of linguistic components which are directly parallel to those needed to analyse and understand a spoken utterance by the user.



Tab. 1 Parallel-like relations intervening between TTS and ASR

As can be gathered from Table 1, there is an almost complete parallelism between the two domains which cannot be regarded "Reversible" in the same way in which a computational system for text understanding cannot be regarded completely reversible vis-à-vis a system for NLP generation (but see 6 - T.Strzalkowski (ed.)).
In particular, a TTS system achieves a good performance by using diphone units derived from a single speech database produced by a single user; on the contrary, an ASR system has to be able to adapt to an undefined number of speakers and the performance may vary according to idiosyncratic traits of the each individual speaker.

Current successful TTS systems go through a choice of the most appropriate diphone unit or speech segment by means of statistical measurements based on a speech segments database collected and annotated in advance for that purpose. Here appropriate means not only containing the legal inventory of phonetic segments, or phones, of the language with all its "most relevant" co-articulatory phenomena; but also reflecting "most" prosodic features of the language/domain - "all" would be impossible due to the number of segments required in order to reach statistical significance.
This is a first important difference between the two parallel domains: ASR can easily attain statistically good results even in a large vocabulary domain due mainly to the fact that the number of phonemes of the human languages is always very limited, and what's more that their contextual realization does not undergo dramatic changes which can be reasonably encoded.
On the contrary, the quantity of information needed to achieve similar performances in TTS is simply too much to be realistically available due to the "sparseness" problem: too many variables have to be taken into account in order to produce a sensible mapping of all linguistically significant and perceptually relevant parameters to reach comparable statistical results in terms of naturalness of speech output.
However, current corpus-based TTS systems [3, 5] tend to approximate the best possible selection of acoustic units on the basis of greedy algorithms, an optimal selection of text to be used as reference corpus by the speakers and a model-based greedy selection of units. By means of such an approach prosodic information is preserved and reproduced to the best possible approximation.
As said at the beginning of this section, the method chosen by most TTS is the concatenative one which is intrinsically incapable of supporting different speech modalities related to the rendering of emotions, or simply to relate the choice of the speech units to one communicative function rather than a generic indication of mood - declarative vs. interrogative.
This notwithstanding, TTS is useful for our Tutoring systems as will be discussed at length in the following sections.

III. REASONING ON MISTAKES FOR FEEDBACK GENERATION

A system for CALL that is aimed at testing the performance of students in text understanding tasks should be equipped with an appropriate feedback to allow for an explanation of the mistakes made. However most systems today replicate the traditional attitude towards feedback: an answer is either right or wrong and no explanation is made available to the student. Drills for text understanding on the computer are usually of two types: multiple choice and true/false choice as will be shown futher on in the paper. Producing free text - even as short texts - as answers to questions is hard to elaborate.
In this section, we present a first approach to such a goal, by describing a system for text understanding GETARUNS, that is used both for text generation and for query-answering. This system is currently being ported to SLIM, the System for Interactive Multimedia Language Learning which has drills both for spoken and written language. The aim is to enable the self-instructional system with a comprehensive feedback system to help the student when working on written and oral drills. The idea described is to use reasoning to build short messages with appropriate feedback to the student when a mistake is detected by the Supervisor which allow SLIM to interact with GETARUN.

SLIM[3, 4] is the prototype interactive multimedia self-learning linguistic software for foreign language students at beginner - false beginner level used in this paper to present TTS in an instructional environment. It allows students to work both in an autonomous self-directed mode or in a way of programmed learning in which the process of self-instruction is preprogrammed and monitored. In this latter mode it is supervised by an Automatic Tutor. Audiovisual materials are partially taken from commercially available courses: all words and utterances of the course have been classified in the

Linguistic Knowledge Database both in orthographic and phonetic form, from all possible linguistic aspects. The most outstanding feature of SLIM is the use of speech analysis and recognition which is a fundamental aspect of all second language learning programmes. We also assume that a learning model is the outcome of the interaction between Student Model and Language Tutor where the former embodies Learning Goals and the latter Pedagogical and Linguistic Knowledge.

Generally speaking, assessment in self-instructional courses is problematic but very important. Self-assessment can be used for appropriate testing purposes - to provide feedback information, diagnostic testing, and placement testing. Within learner-centred self-instruction, or self-directed learning, self-assessment is a necessary part. Decisions about whether to go on to the next item, exercise or unit, decisions concerned with the allocation of time to various skills, or with the need of remedial work, are all based on feedback from informal and formal assessment. This concept then is central both to the learners' personality and to the kind of courseware we are building. We consider it important as an educational goal in its own right, and training learners in this is beneficial to learning.

In fact, language learners regularly engage in self-assessment as part of their learning. They make exercises and check, by whatever means available, whether their responses are correct or not. They check the computer's comprehension of their spoken language, and adjust it when necessary. In a language like English, the ability to perform a complete phoneme-to-grapheme translation in L2 is severely undermined by its phonotactics which is full of exception and requires a lot of exercise to couple understanding and orthographic abilities. As for written language skills the number of assessment tools is fairly extended and are based essentially on the knowledge the computer has of every single linguistic item considered in a given task. For instance, in case the system is assessing the learners' achievements in grammatical knowledge, it accesses the Linguistic Knowledge Database (hence LKD) where each item may correspond either to a word-form, or a syntactic phrase for syntactic tests; or still to an utterance for context-based pragmatic and communicative function tests.

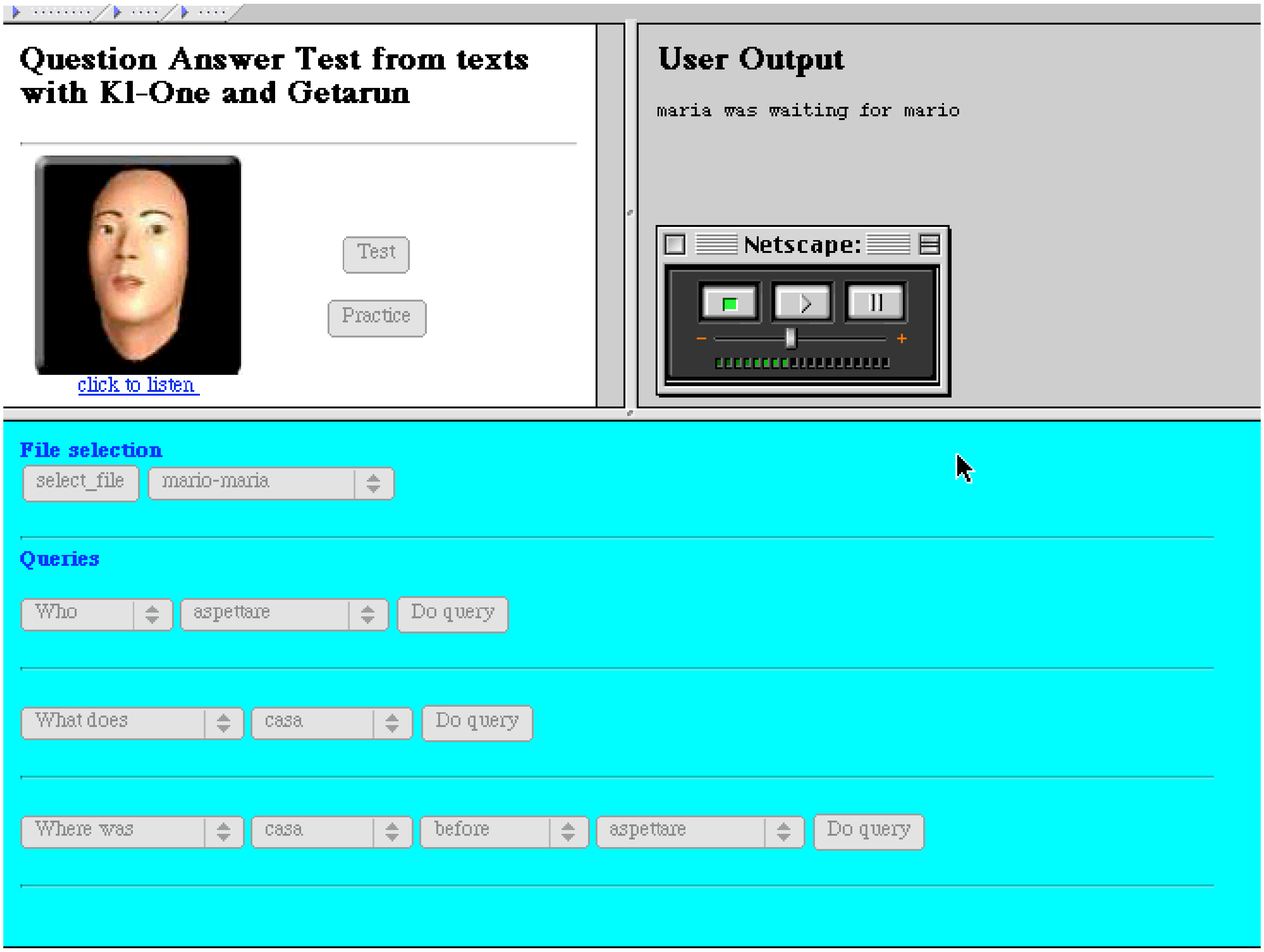
The LKD is the foundation for all drills construction, and thus it constitutes the basis of all self-assessment activities. In particular, the AT may create an infinite number of drills automatically since it has been given an internal pedagogical and linguistic set of criteria on the basis of which it may choose at random from the LKD the items relevant and adequate for any given linguistic task.

The AT has also been equipped with a number of tools that enable it to check and spot mistakes and errors whenever they are made by the learner, and keep record of them. Errors may be noted in both oral and written activities, and will be simply notified to the student in Free Modality or communicated by the Supervisor when working in Guided Modality.

In all cases, learners will be informed about the error, the kind of error they produced, the possible reason why they made that kind of error: as a side-effect, they will be directed to carry out some linguistic activity appropriate to help remedy that problem or else the grammar section on that item will be shown. The same will happen with phonetic problems: in case the performance scores too low, the phonetics hypertext will be called and presented to the student at the appropriate item.

However, text understanding tasks constitue a challenge in that the right feedback may not be available if the student provides an incorrect answer not included amongst the list of possible mistakes. We use this modality with listening comprehension tasks in which the student is read a text aloud by the internal Text-To-Speech module or by a previously recorded text, and there is no written text available. At the end of the listening activity, a certain number of questions will appear on the screen and the student will be prompted to provide answers to each one.

In Fig.1 below we show the Activity Window in Internet for Text Understanding. The student may choose between doing Practice and doing Test activities: in the former case, the choice of text will be with the student who will also be able to listen to the spoken synthetic text as many times as he may need. After Starting Prolog and Selecting a text, the student will be allowed to ask as many queries he likes on its content in order to facilitate a full understanding. When the student is ready and Test Activity is chosen, the system will choose randomly one text and propose it to the student in the spoken modality, twice. After that the student will have to build adequate statements on the contents of the story he heard by activating a number of menus on the lower part of the screen. These menus contain all events, properties and participants in the story with the addition of a certain number of intruders which are phonetically close the actual linguistic items used in the story. We expect the student to make two types of mistakes:



Question Answer Test from texts with Kl-One and Getarun

User Output

```
maria was waiting for mario
```

Fig. 1 Text Understanding Activities under Internet with KL-One and GETARUN

a. the words chosen are not part of the text making up the story.
b. the words chosen are all part of the text making up the story, however the statement built by the student is wrong and does not reflect the contents of the story.

Feedback in the former case will indicate clearly that the student has misunderstood and/or simply not understood part of the text heard. In the latter case, on the contrary, his proficiency of the language is not such that he may execute the exercise appropriately: this may be either due to a misunderstanding less bad than the one detected in the former case; else, depending on the type of error made, he may be in need of more practice. This will be discussed in more detail below: we shall now present the reasoning part of the system.

GETARUN[1, 2, 5] is a system for text and reference understanding which is currently used for summarization and text generation. It has a highly sophisticated linguistically based semantic module which is used to build up the Discourse Model. Semantic processing - see Tab.1 - is strongly modularized and distributed amongst a number of different submodules which take care of Spatio-Temporal Reasoning, Discourse Level Anaphora Resolution, and other subsidiary processes like Topic Hierarchy which will impinge upon Relevance Scoring when creating semantic individuals. The Knowledge Representation Language we use is an offspring of KL-ONE[7]. We are currently using Version 5.2 released in September 1993. Functionality of the system are reasoning based on terminological logics, supporting inheritance, consistency checking, cycle detection, classification, completion of partial descriptions, role inferences, A-box revision, extended query answering.

The main system components are: a TBox containing the KB scheme, an ABox containing the KB assertions, the IBox containing the extensional implications. System interfaces are constituted by an uniform access language that will be briefly exemplified below. Input to the reasoning system is the output of GETARUN, in the form of a Discourse Model.

3.1 The Discourse Model or DM

Informally, a DM may be described as the set of entities "naturally evoked" [13] by a discourse, linked together by the relations they participate in. They are called discourse entities, but may also be regarded as discourse referents or cognitive elements. We want to keep referring to what people do with language; evoking and accessing discourse entities are what texts/discourses do. A discourse entity inhabits a speaker's discourse model and represents something the speaker has referred to. A speaker refers to something by utterances that either evoke (if first reference) or access (if subsequent reference) its corresponding discourse entity. It is how the information is realized that determines what types of discourse entities are available when.

Now, a speaker is usually not able to communicate all at once the relevant properties and relations he may want to ascribe to the referent of a discourse entity. To do that he may have to direct the listener's attention to that referent (via its corresponding discourse entity) several times in succession. When the speaker wants to re-access an entity already in his DM (or another one directly inferable from it), he may do so with a definite anaphor (pronoun or NP). In so doing, the speaker assumes (1) that on the basis of the discourse thus far, a similar entity will be in (or directly inferable from) the listener's growing DM and (2) that the listener will be able to re-access (or infer) that entity on the basis of the speaker's cues. (For example, pronouns are less of a cue that anaphoric NPs). The problem then, at least for definite anaphora, is identifying what discourse entities a text naturally evokes.

What characterizes a discourse entity? Webber's view is that a discourse entity is a "conceptual coathook" (a term coined by William Woods) on which to hang descriptions of the entity's real world or hypothetical world correspondent. As soon as a DE is evoked, it gets a description. Over the course of the text, the descriptions it receives are derived from both the content of the speaker's utterances and their position within the discourse, as well as whatever general or specific information about the discourse entity the listener can bring to bear.

The initial description ID that tags a newly evoked DE might have a special status, because it is the only information about an entity that can, from the first and without question, be assumed to be shared (though not necessarily believed) by both speaker and listener alike.

Even though certain types of DE must be derived from other ones inferentially, and that is the simplest way of account for anaphoric access to "generic set" DE.

The problem we set out to solve is transformed into

c. identifying the discourse entities a text evokes and

d. ascribing to them appropriate IDs;

associating relations and properties to each ID.

Each text given to the student is kept in the form of a Discourse Model and turned into an appropriate database structure by KL-One which provides descriptive knowledge to allow for reasoning to take place. We shall comment the system's behaviour on the basis of the following short text:"At the Restaurant".

"John went into a restaurant. There was a table in the corner. The waiter took the order. The atmosphere was warm and quiet. He began to read his book."

For instance, with a sentence like "John went into a restaurant" the DM will look like this:

DISCOURSE MODEL

```
sentence(r01.new,[john, went, into, a, restaurant])
loc(infon3, id1, [arg:main_tloc, arg:tr(f5_r01)])
loc(infon4, id2, [arg:main_sloc, arg:restaurant])
ind(infon5, id3)
fact(infon6, inst_of, [ind:id3, class:man], 1, univ, univ)
fact(infon7, name, [john, id3], 1, univ, univ)
fact(infon9, isa, [arg:id2, arg:restaurant], 1, id1, id2)
fact(id4, go, [agente:id3, locat:id2], 1, tes(f5_r01), id2)
fact(infon10, isa, [arg:id4, arg:ev], 1, tes(f5_r01), id2)
fact(infon11, isa, [arg:id5, arg:tloc], 1, tes(f5_r01), id2)
fact(infon12, past, [arg:id5], 1, tes(f5_r01), id2)
overlap(tes(f5_r01), td(f5_r01))
sentence(r02.new, [there, was, a, table, in, the, corner])
ind(infon21, id6)
ind(infon22, id7)
fact(infon23, inst_of, [ind:id7, class:thing], 1, univ, univ)
fact(infon24, isa, [ind:id7, class:corner], 1, id1, id2)
fact(infon25, in, [arg:id6, locativo:id7], 1, id1, id2)
fact(infon26, isa, [ind:id6, class:table], 1, id1, id2)
fact(infon27, inst_of, [ind:id6, class:thing], 1, univ, univ)
fact(id8, there_be, [tema_nonaff:id6], 1, tes(f4_free_r02), id2)
fact(infon31, isa, [arg:id8, arg:st], 1, tes(f4_free_r02), id2)
fact(infon32, isa, [arg:id9, arg:tloc], 1, tes(f4_free_r02), id2)
fact(infon33, past, [arg:id9], 1, tes(f4_free_r02), id2)
included(tr(f4_free_r02), id1)
contains(tes(f4_free_r02), tes(f5_r01))
sentence(r03.new, [the, waiter, took, the, order, .])
ind(infon42, id10)
fact(infon43, inst_of, [ind:id10, class:[social_role]], 1, univ, univ)
fact(infon44, isa, [ind:id10, class:waiter], 1, id1, id2)
fact(infon45, role, [waiter, id2, id10], 1, id1, id2)
fact(id12, take_order, [actor:id10, goal:id3], 1, tes(f2_free_aq), id2)
fact(infon48, isa, [arg:id12, arg:pr], 1, tes(f2_free_aq), id2)
fact(infon49, isa, [arg:id13, arg:tloc], 1, tes(f2_free_aq), id2)
fact(infon50, past, [arg:id13], 1, tes(f2_free_aq), id2)
included(tr(f2_free_aq), id1)
after(tes(f2_free_aq), tes(f5_r01))
sentence(r04.new, [the, atmosphere, was, warm, and, clear])
loc(infon59, id14, [arg:main_tloc, arg:tes(f2_free_aq)])
fact(infon60, isa, [arg:id15, arg:air], 1, id14, id2)
fact(infon61, [clean, nice], [arg:id15], 1, id14, id2)
fact(id16, be, [prop:infon61], 1, tes(f5_r04), id2)
fact(infon62, isa, [arg:id16, arg:st], 1, tes(f5_r04), id2)
fact(infon63, isa, [arg:id17, arg:tloc], 1, tes(f5_r04), id2)
fact(infon64, past, [arg:id17], 1, tes(f5_r04), id2)
included(tr(f5_r04), id14)
contains(tes(f5_r04), tes(f2_free_aq))
```

```
sentence(r05.new, [he, began, to, read, his, book])
fact(infon76, isa, [arg:id18, arg:book], 1, id14, id2)
fact(infon77, poss, [arg:id18, poss:id3], 1, id14, id2)
fact(id19, read, [agente:id3, tema_aff:id18], 1, tes(finf1_free_a1), id2)
fact(infon78, isa, [arg:id19, arg:pr], 1, tes(finf1_free_a1), id2)
fact(infon79, isa, [arg:id20, arg:tloc], 1, tes(finf1_free_a1), id2)
fact(infon80, pres, [arg:id20], 1, tes(finf1_free_a1), id2)
fact(id21, begin, [agente:id3, prop:id19], 1, tes(f3_free_a1), id2)
fact(infon82, isa, [arg:id21, arg:ev], 1, tes(f3_free_a1), id2)
fact(infon83, isa, [arg:id22, arg:tloc], 1, tes(f3_free_a1), id2)
fact(infon84, past, [arg:id22], 1, tes(f3_free_a1), id2)
included(tr(f3_free_a1), id14)
after(tes(f3_free_a1), tes(f2_free_aq))
```

3.2 MAPPING SEMANTIC REPRESENTATIONS INTO THE KNOWLEDGE BASE

Conceptual Representations(CR) have been introduced by Jackendoff [8,9,10,11,12] who introduced a number of augmentation to the original set which we also endorse. In [1] CRs were considered the link from the semantics to the knowledge of the world needed to represent meaning in a general and uniform manner. The Discourse Model only contains reference to semantic roles and other semantic relations like Poss, which have a correspondence in the CR. Here below are CRs for some of the verb predicates of the text under analysis.

```
pred(exist[BE(<theme_unaffect>(STAYposit(AT)))          {(en,tn),(e1,t1)}])
pred(enter[CAUSE(<agent>(GOposit(FROM[x]     (INTO<locat_into>)))){(en,tn),(e1,t1)}])
pred(take_order[CAUSE(<actor>(GO(FROM<goal>)))     {(e1,t1),(en,tn)}])
pred(read[LET(<address>(GO(REP(FROM<informtn>))){(en,tn),(e1,t1)}])
```

The encoding of relations and properties is done by means of CRs and a certain number of additional inference rules like the following:

```
a) [CAUSE (X,E) at t1] => [E] cond = +specific(t1)
b) [STAY ([X],[AT Y]) from t1 to t2] =>
        [BE ([X],[AT Y]) at t3] cond = t1<t3<t2
c) [GO([X],[FROM Y],[TO Z]) at t1] =>
        [BE ([X],[AT Y]) at t2] &
        [BE ([X]?[AT Z]) at t3]
                cond = t2<t1<t3
```

The reading of these expressions is quite intuitive: in a) if an agent X causes E than E takes place, under the condition that reference time be specific; b) is the subinterval condition which is cast into the formalism for temporal reasoning; c) shows how a motion predicate is translated into a couple of state predicates and so on.

```
d) [GO ([X],[(AWAY_)FROM Y],[TO(WARD) Z]
        from t1, to t2] =>
        NOT [STAY ([X],[AT Y])
        from t1, to t2] & NOT [STAY ([X],
             [AT Z]) from t1, to t2]
e) [STAY ([X],[AT Y] from t1, to t2] =>
        NOT [GO ([X],[(AWAY_)FROM Y], [TO(WARD)W]) from t3, to t4]
        cond = t1<t3<t4<t2
```

3.3 Definition of sets, general class, social roles and time intervals.

At first, very general roles and properties are declared, like for instance the set of proper names or the set of prepositions:

We then define attributes to be associated to individuals which we recover from the DM and divide up accordingly into three main general group: PERSONS, OBJECTS and PLACES.

Aspectual information is used to individuate the appropriate internal constituency of the event and also to drive the semantics, which together with the information coming from arguments and adjuncts will be able to trigger the adequate knowledge representation. In particular, we need to process reference to entities and events in the discourse model, in order to know what predicates are asserted to hold over what entities and when.

Finally, EVENTS, STATES and PROCESSES are special objects which possess a polarity, a reference time and an agent that causes something: it may be a process as in the case of BEGIN, or and event of going to a place as for GO, or still a theme_nonaff (a non affected theme that stays in a given location.

```
events  :- ev        :< general
 and all(time,tense) and all(polarity,pol),
 specify_event_predicates.
states  :- st      :< general
and all(time,tense) and all(polarity,pol),
specify_state_predicates.
specify_event_predicates:-
fact(_,isa,[arg:X,arg:ev],1,A,B),
fact(X,Pred,[Agent:_,locat:_],1,B,_),
Pred :< ev
and all(agent,man) and all(location,place)
     and all(prim,primCause) .
specify_state_predicates:-
fact(_,isa,[arg:X,arg:st],1,A,_),
fact(X,Pred,Arguments,1,B,_),
Pred :< st
and all(theme_nonaff,thing) and all(prim,primbe).
```

From general entities we then declare instances as they have been collected in the DM with their semantic ids, for instance for all instances of man we use,

```
facts :- fact(_,inst_of,[ind:Y,class:man],1,_,_),
          Y :: man.
```

For all instances of individuals belonging to the class of social_roles,

```
facts :- fact(_,inst_of,[ind:Y,class:social_role],1,_,_),
        fact(_, isa, [ind:Y, class:Class], 1, _, _),
          Y :: Class.
```

Names and roles are associated to a specific relation:

```
facts :- fact(_,name,[X,Y],1,_,_),
          Y :: has_name:X.
facts :- fact(_,role,[Class,X,Y],1,_,_),
          Y :: works_in:X.
```

The same applies for ids associated to spatiotemporal locations, both for entities and relations like events, processes or states,

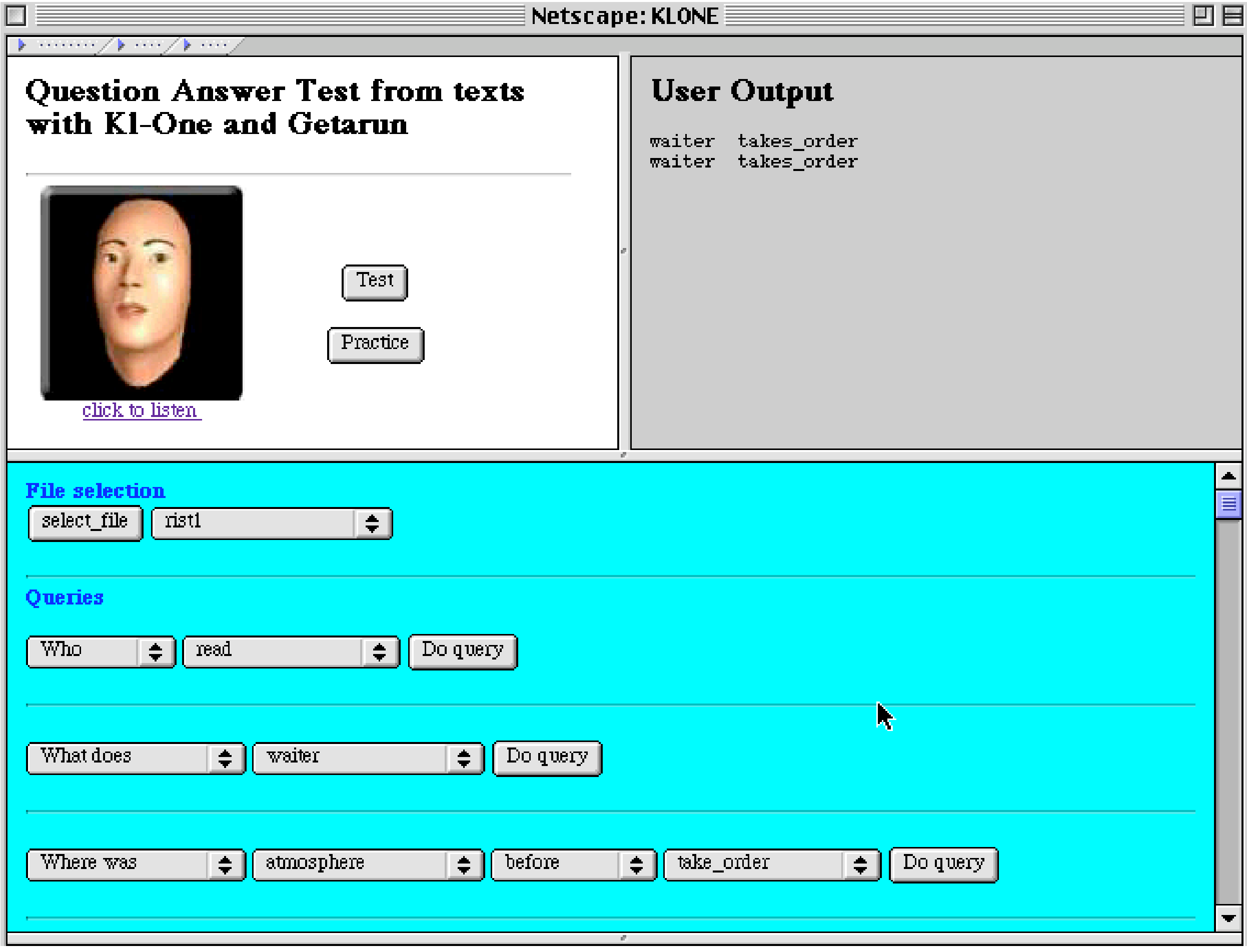
```
facts :- fact(_,isa,[ind:X,class:Y],1,A,Z),
          X :: Y and time:A and location:Z.
facts :- fact(_,isa,[arg:X,arg:ev],1,A,B),
```

X :: ev and time:A and location:B.

Then we associate to a relation id all its semantic properties, from semantic roles to spatiotemporal relations, as for instance with predicates like GO,

facts :- fact(X,Pred,[agent:Z,locat:A],1,B,_),

X :: Pred and agent:Z and location:A and time:B.



Question Answer Test from texts with Kl-One and Getarun

User Output

```
waiter  takes_order
waiter  takes_order
```

File selection

Queries

3.4 SEMANTIC Ids & QUERIES TO THE SYSTEM

Here below are some of the queries that can be addressed to the system. We propose the canned sentence proposed to the user, then the internal output of the query, and finally the answer generated by the internal generator.

"Where was John?"

| ?- where_was(id3).

After tes(f5_r01) was in id2

after entering john was in the restaurant

And similarly we may ask the following questions and get the appropriate answer.

"Where was the table?"

" Where was john after being entered "

"Where was the waiter?"

"Who took the order?"

"Who ordered?"

"Who was reading the book?"

“What was in the restaurant?”

The interesting part of the program is obviously the possibility of recovery from failure in case of wrong inferences. According to the type of query, failure may be recovered and an appropriate feedback generated. For instance, in case the answer to the question,

“Who was reading the book?”

is "the waiter", the feedback generated could be,

"No, the waiter works in the restaurant: John is reading the book!"

Similar recovery strategies can be easily set up simply by backretrieving information related to the wrong answer and then generate a message which is made up of two parts: an explanation of the error in a first sentence and the right answer in the second sentence.

IV. TTS and the AUTOMATIC DICTATION EXERCISE

The simplest way of introducing TTS in a CALL courseware is to use it for dictation exercises. TTS can read any text, especially if new texts need to be introduced frequently and there is no possibility to have it appropriately recorded by a native speaker. In addition, if we think of Dictation as a supporting activity in a regular course, or just as a test, it needs the recording to be done appropriately as if the speaker had a student or a class to talk to in front of him. In certain cases, reading speed must be adjusted to make a certain portion of text more understandable and/or it must be read twice or even more times, depending on level of proficiency of practicing students.

All these specific requirements are perfectly satisfiable by a computerized version of the Dictation Exercise and we shall present one such implementation below. In addition, the use of TTS allows different voices, be it male or female, as well as different accents for English or for other languages such as Spanish.

Implementing a Dictation exercise is done with a pre-editing phase in which the input text is appropriately formatted in order to have it split at the right speaking interval: it is broken up into fragments, each one corresponding to a paragraph, and all punctuation is spelled out to allow for it to be read aloud, and used as first approximation to the problem of chunking. Chunking is usually done by the human tutor almost automatically, by simply going through the written text and by indicating positions where a pause should be introduced with a slash. However, this cannot be dubbed automatically by the computer application, and needs pre-editing, i.e. it needs a specific indication of the pause to be inserted in the text to be read by the TTS. For this reason, there must be a symbol to be used as a pause, and we decided to choose the empty list, []. Here below we show a text by E.A.Poe semi-automatically pre-edited to serve for the system:

> from Edgar Allan Poe,
> THE OVAL PORTRAIT
> •
>
> We saw a beautiful house many hundreds of years old.[full stop] My servant Pedro took me in.[full stop] I was badly hurt and I would die if we stayed out all night.[full stop] All the rooms were decorated with many fine pictures.[full stop] I couldn't sleep because of the pain.[full stop] So I looked at the pictures,[comma] and read a small book that described all the pictures.[full stop] I put a lamp closer to the wall and saw an oval picture of a very beautiful young woman.[full stop]
>
> I then closed my eyes.[full stop] It was the finest painting I had ever seen.[full stop] I stayed for an hour looking at the girl;[semicolon] the more I looked the more I became afraid.[full stop] So I read the story of the picture.[full stop] She married the painter and was very happy.[full stop] Sadly she soon saw he was already married:[colon] to his

> work,[comma] and his painting seemed so important.[full stop] Slowly she started to dislike her husband's work.[full stop] She felt terrible one day when he announced that he wanted to paint her.[full stop]
> He occupied all his time —[dash] day and night —[dash] to complete this portrait of his love.[full stop] He almost never left his paints.[full stop] He wanted it perfect.[full stop] But he couldn't see that slowly she was getting weaker every day.[full stop] Her face was now white as snow.[full stop] He no longer was painting what was in front of him [] but was working on a picture he was in love with.[full stop]
> When he finished his work in the middle of winter,[comma] he stoo back and started to shake with happiness and fear.[full stop] All his colour left his face.[full stop] He cried,[comma] "[open inverted commas] This woman is not made of paint![exclamation mark] She is alive".[close inverted commas & full stop] Then he turned to his wife that he loved so much,[comma] and saw she was now dead.[full stop]

I said semi-automatically simply because "inverted commas" command cannot be automatically encoded and needs the pre-editor to search for the open/close pair manually. As to the remaining punctuation commands they can all be uniquely identified, and simply translated into the corresponding orthographic expressions, which, as should be clear now, also correspond to chunks boundaries and to pauses for the TTS:

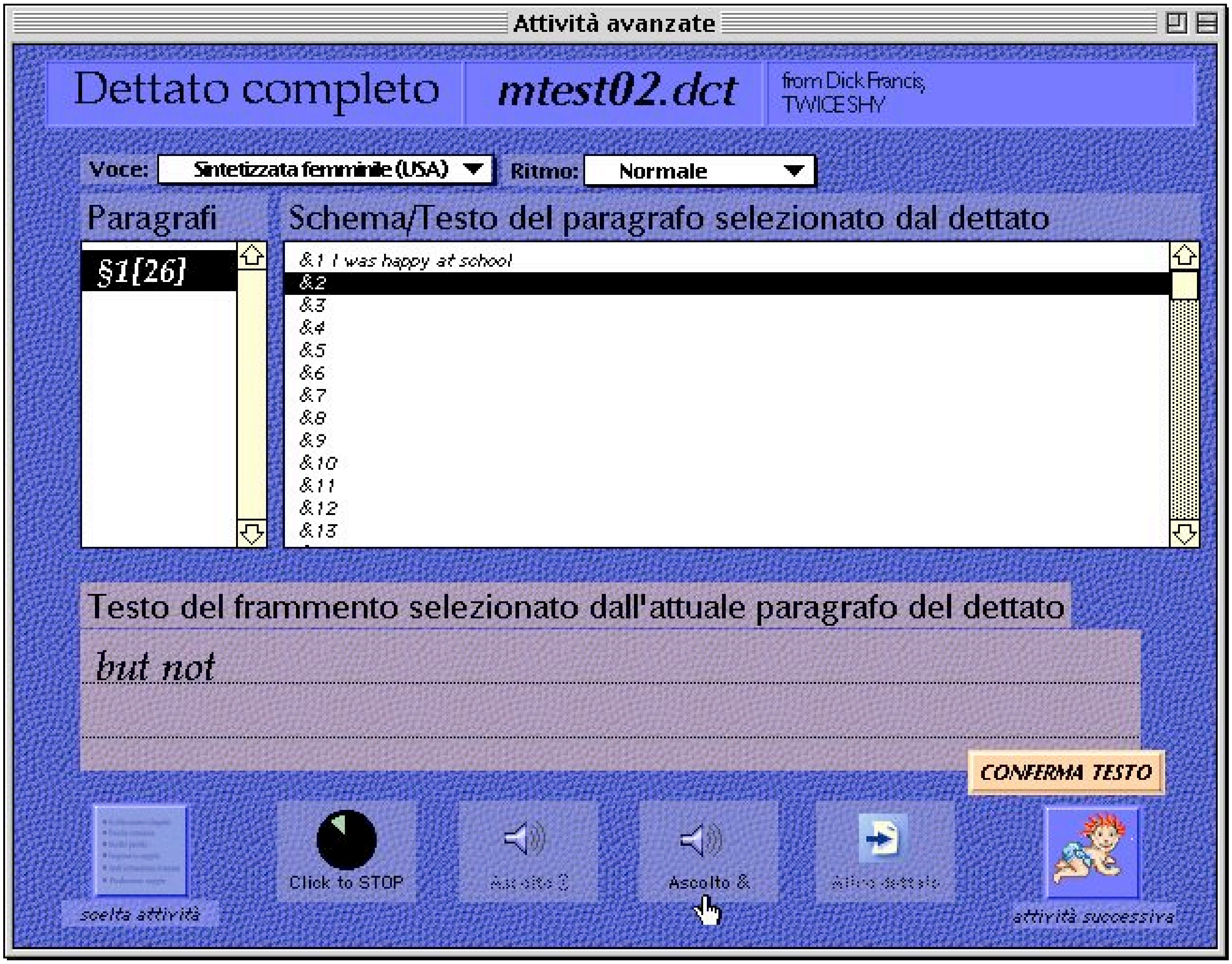
e. .[full stop]
f. ,[comma]
g. —[dash]
h. ![exclamation mark]
i. :[colon]
j. ?[question mark]
k. ;[semicolon]
l. "[open inverted commas]
m. "[close inverted commas]
n. [] induced pause

The induced pause command, represented by the empty list, is needed either because there is no explicit indication for a pause by punctuation in the original text.

When reading a text for dictation purposes, the human/automatic tutor has to obey to a number of important constraints:

o. chunking the text for semantic units;
p. chunking the text for breath units corresponding to comprehensible and retainable portions of texts;
q. allowing sufficient time to the student for the text to be transferred into orthographic form;
r. all the exercise must be completed within a given stretch of time, usually half an hour - and the same will have to be allowed to the student working with the courseware;
s. repeating the chunk as many times as required by certain hard to process chunks, or as stipulated at the start - every chunk is usually read twice;
t. in addition - and this may only apply to the computer application - reading speed may be automatically set to faster or slower by changing speaking rate of the TTS device;
u. finally, the human tutor will have to produce a summary of mistakes after correcting each student's paper.

All this is done automatically and in real time by the system. In particular, checking for misspelled words will be done after the student has decided that the text he/she has written corresponds to what he/she has heard from the synthetic voice, and has confirmed that by pushing the button on the low right-hand side of the written input section of the Activity Window for Complete Dictation shown here below.



Attività avanzate

Dettato completo

mtest02.dct

from Dick Francis,
TWICE SHY

Voce: Sintetizzata femminile (USA)

Ritmo: Normale

Paragrafi

§1[26]

Schema/Testo del paragrafo selezionato dal dettato

&1 I was happy at school
&2
&3
&4
&5
&6
&7
&8
&9
&10
&11
&12
&13

Testo del frammento selezionato dall'attuale paragrafo del dettato

but not

Tab.1 Activity Window for Dictation Exercise in SLIM courseware

I shall now comment in detail the contents of the Activity Window for Dictation Exercise in SLIM. In the higher portion of the Window there is the current file name and the author's name associated with the text to be dictated. Below there are two menus: one allowing to choose one voice type for the TTS amongst a number of different voices available on a Macintosh computer, and the other menu allows the student to change speaking rate and consequently the speed with which the text will be read. The central portion of the Window has two scrolling spaces: one on the left associated with paragraph numbering and one on the right associated with Chunks. Each chunk will be given a separate check by the system when comparing the student's with the original written text and looking for mistakes. In addition, whenever one or more words have been misspelled, the system will show them as crossed missing elements in the text sequence of the current chunk.

In the portion of the screen below the input writing section there is a sequence of buttons allow the student to move out of the current Activity either going back to the main screen for the Writing

Activities Package - first button on the left - or by moving on to the next Activity, the last button on the right. The four central buttons are organized as follows:

v. first button is a clock that tells the student how much time left he/she has got to complete writing the dictation for the current Paragraph;
w. second button allows the student to listen to the text of the complete Paragraph;
x. third button allows the student to listen to the text of the current Chunk;
y. fourth button allows the student to move onto the next Paragraph.

V. TTS and TEXT COMPREHENSION

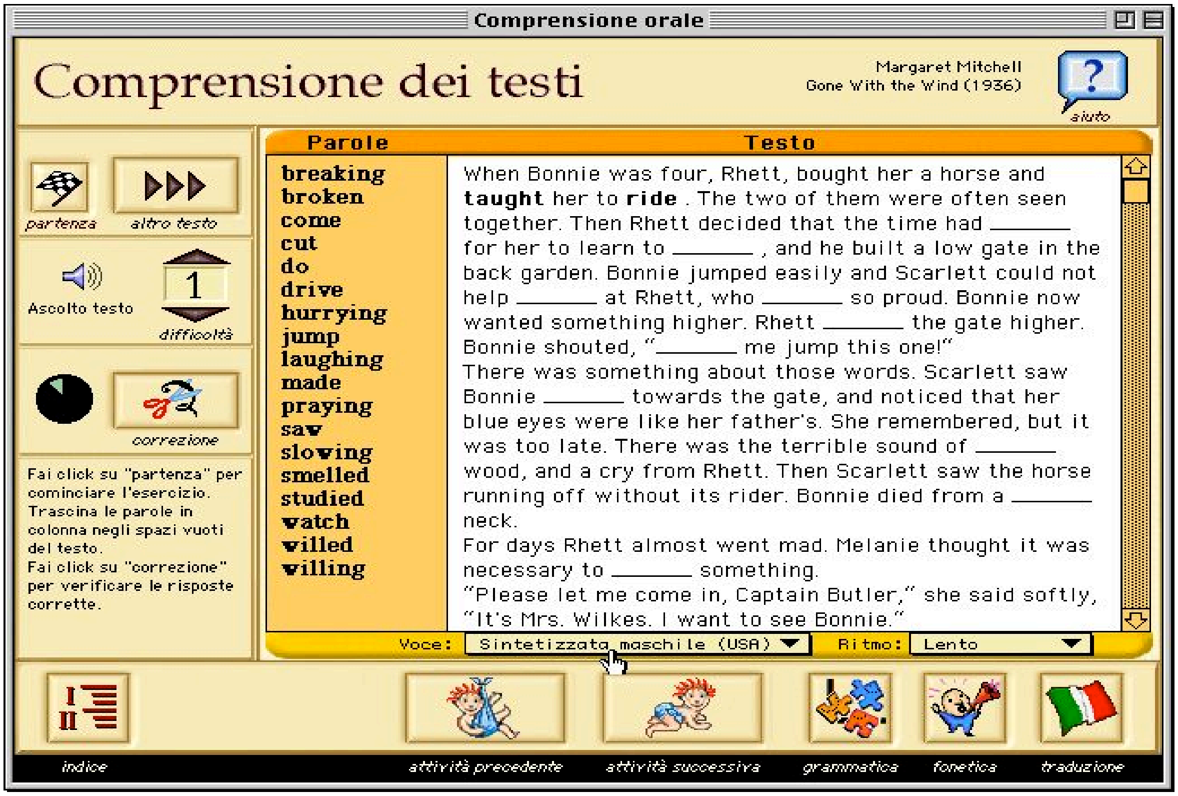
A number of less complicated exercises can be organized in which TTS can be used profitably: we shall comment on two of them, Text Comprehension and Cloze Test. The former exercise requires the student to listen to a text read aloud by the system with no special intervention on the text itself. The student may listen to the text read a certain number of times. At the end of the listening comprehension the student will be presented with a number of True/False statements on the text to be validated. In a Beginners' version also the written text will be presented and the student will be allowed to browse it for a limited amount of time before answering.

A second exercise requires the student to fill-in slots left empty in a Cloze Test: listening to the read text will help the student to carry out the task in case the proficiency level is lower than Advanced, where this facility is not available.

In a third exercise, the student is presented with a sentence in which words have been scrambled by the system: he/she will be allowed to listen to the read version of the reordered sentence again according to its proficiency level. The student will then have either to reorder the sentence manually by positioning each word in the right sequence, or - in case he/she is an Advanced student - speak the sentence to the ASR engine which will take care of recognizing whether the sentence has been rightly or wrongly reorganized.

We show here below three screen snapshots for the exercises: the Text Listening Comprehension, the Cloze Test and Sentence Reordering. The former Activity Window is very simple: it has a text scrolling portion in the middle which appears for a short time after the text has been completely read aloud; below this central portion there are the True/False statements to be validated by the student. On the left there is a Start button and a Choose another Text above and below a Speak button. Below that, there is a Clock and a Correction button.

The second snapshot is related to the Cloze Test which is very similar to the previous one. The only difference is in the presence of a list of Words to be inserted in the empty slots left by the Cloze.



Margaret Mitchell
Gone With the Wind (1936)

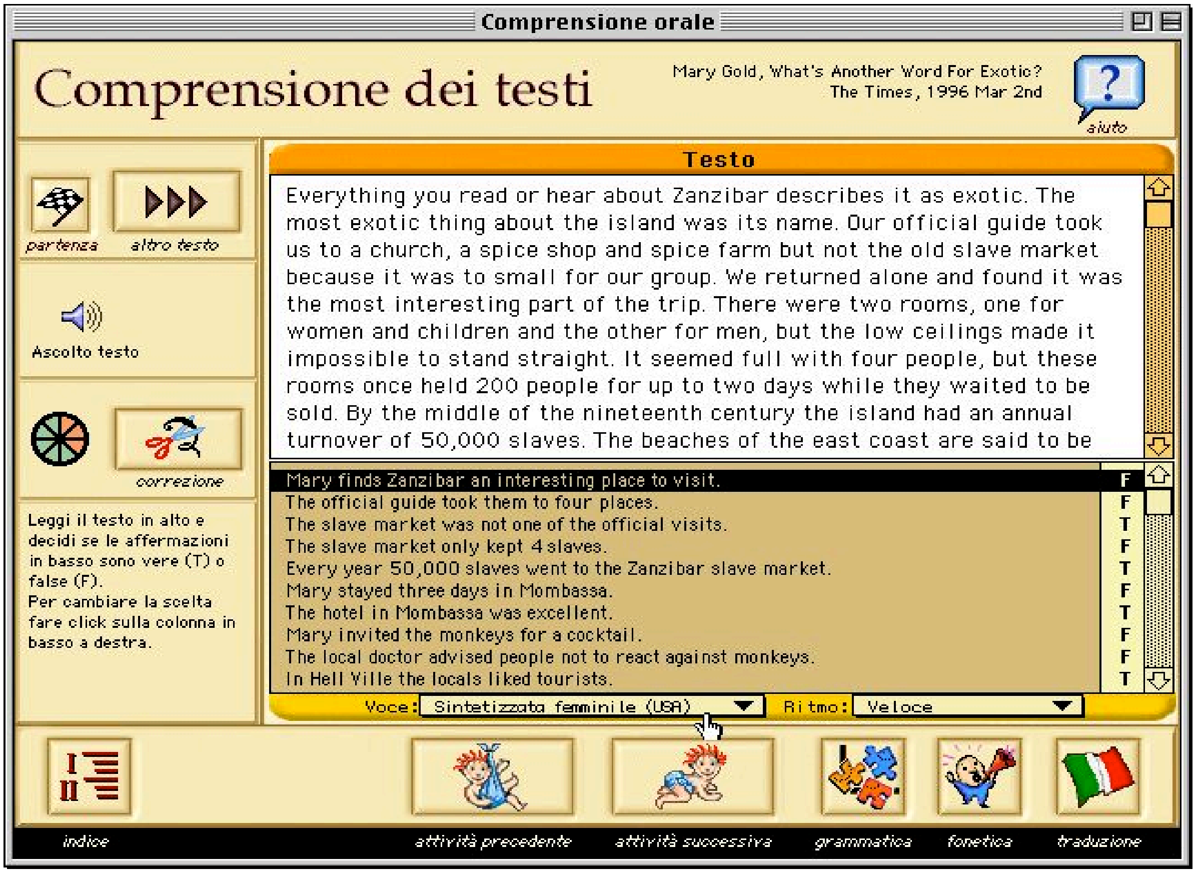
Fai click su "partenza" per cominciare l'esercizio. Trascina le parole in colonna negli spazi vuoti del testo. Fai click su "correzione" per verificare le risposte corrette.

breaking
broken
come
cut
do
drive
hurrying
jump
laughing
made
praying
saw
slowing
smelled
studied
watch
willed
willing

When Bonnie was four, Rhett, bought her a horse and **taught** her to **ride** . The two of them were often seen together. Then Rhett decided that the time had _______ for her to learn to _______ , and he built a low gate in the back garden. Bonnie jumped easily and Scarlett could not help _______ at Rhett, who _______ so proud. Bonnie now wanted something higher. Rhett _______ the gate higher. Bonnie shouted, "_______ me jump this one!"
There was something about those words. Scarlett saw Bonnie _______ towards the gate, and noticed that her blue eyes were like her father's. She remembered, but it was too late. There was the terrible sound of _______ wood, and a cry from Rhett. Then Scarlett saw the horse running off without its rider. Bonnie died from a _______ neck.
For days Rhett almost went mad. Melanie thought it was necessary to _______ something.
"Please let me come in, Captain Butler," she said softly, "It's Mrs. Wilkes. I want to see Bonnie."

Voce: Sintetizzata maschile (USA)
Ritmo: Lento

Tab. 2 Activity Window for Listening Text Comprehension



Mary Gold, What's Another Word For Exotic?
The Times, 1996 Mar 2nd

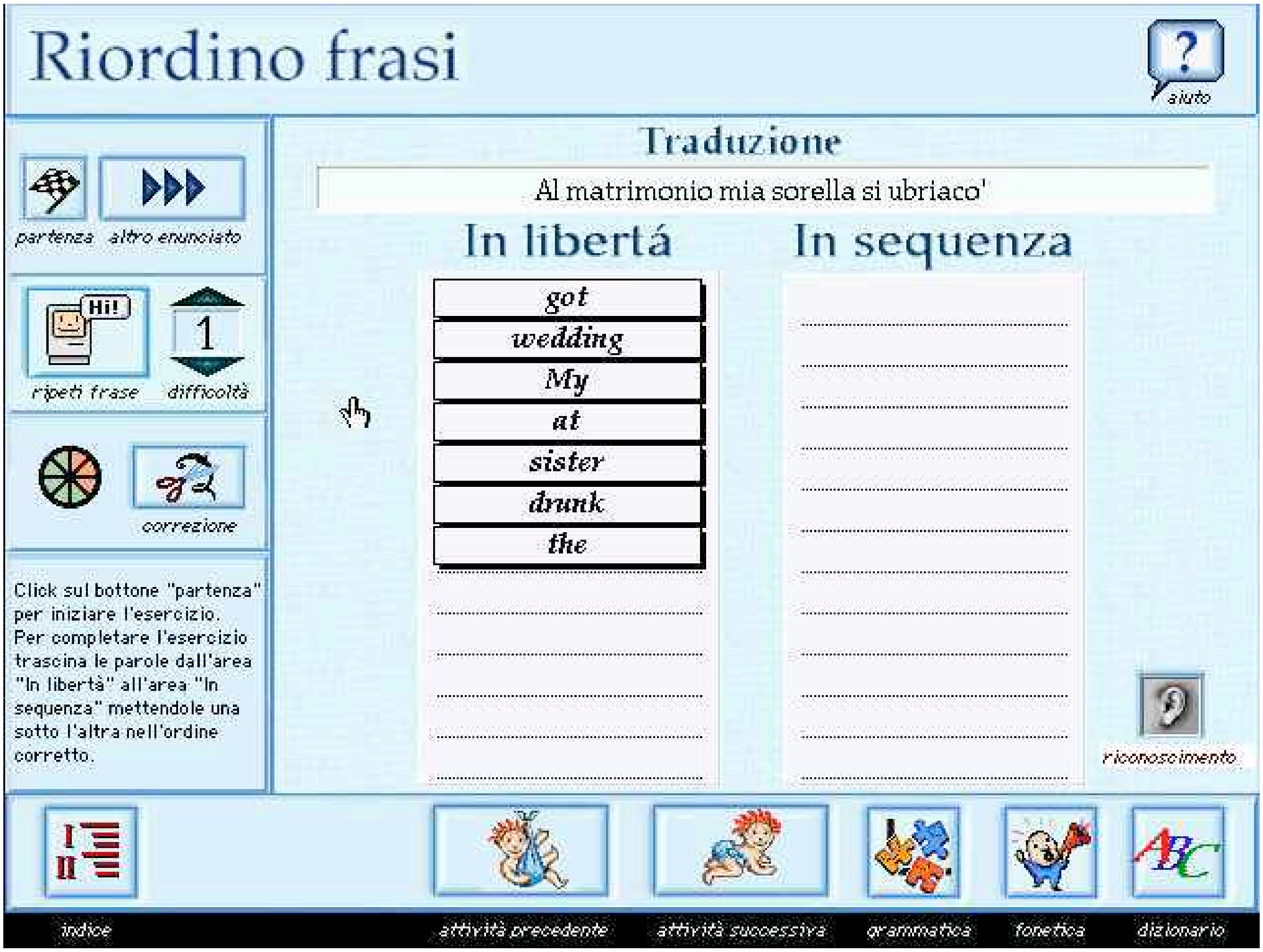
Leggi il testo in alto e decidi se le affermazioni in basso sono vere (T) o false (F).
Per cambiare la scelta fare click sulla colonna in basso a destra.

Everything you read or hear about Zanzibar describes it as exotic. The most exotic thing about the island was its name. Our official guide took us to a church, a spice shop and spice farm but not the old slave market because it was to small for our group. We returned alone and found it was the most interesting part of the trip. There were two rooms, one for women and children and the other for men, but the low ceilings made it impossible to stand straight. It seemed full with four people, but these rooms once held 200 people for up to two days while they waited to be sold. By the middle of the nineteenth century the island had an annual turnover of 50,000 slaves. The beaches of the east coast are said to be

Mary finds Zanzibar an interesting place to visit.	F
The official guide took them to four places.	F
The slave market was not one of the official visits.	T
The slave market only kept 4 slaves.	F
Every year 50,000 slaves went to the Zanzibar slave market.	T
Mary stayed three days in Mombassa.	F
The hotel in Mombassa was excellent.	T
Mary invited the monkeys for a cocktail.	F
The local doctor advised people not to react against monkeys.	F
In Hell Ville the locals liked tourists.	T

Voce: Sintetizzata femminile (USA)
Ritmo: Veloce

Tab. 3 Activity Window for Cloze Test



Riordino frasi

Traduzione

Al matrimonio mia sorella si ubriaco'

In libertá

got
wedding
My
at
sister
drunk
the

In sequenza

Click sul bottone "partenza" per iniziare l'esercizio. Per completare l'esercizio trascina le parole dall'area "In libertà" all'area "In sequenza" mettendole una sotto l'altra nell'ordine corretto.

VI. TTS and LEXICON LEARNING

We shall now introduce the exercise of learning the lexicon of a second language, which in some case may also be a minor language that is difficult to practice [6, 7]. There are two interesting range of problems that TTS might help to solve:

- Offering a first level linguistically appropriate exposure to the target language by presenting the sound system of the language in their context;
- Allowing the student to couple meaning, sound and sometimes corresponding image to the words to be learnt, thus making the task simpler;
- Provide an easy authoring facility by allowing tutors to increase the size of lexicon, by simply adding new words into the database;

z. allowing an interlanguage exposure to the learner and prompting him to compare the different pronunciations according to proficiency level;

- more generally, allowing adaptation to speaking rate and voice type and quality which is usually impossible to obtain with real life recording.

It is a fact that lexical knowledge is the first obstacle a student of a second language has to face in the learning process. CALL devices incorporating TTS may be very helpful simply on the basis of the fact that a multisensory learning environment is much more effective than a lean one dimensional one. In particular then, it is certainly more efficient, let alone more natural, to learn words of a second language by associating sound to orthographic image. It would even improve the learning environment if some adequately preelaborated still image representing the meaning/concept expressed by the word accompanies the sound which usually happens with children. This is much more so, in case the second language to be studied is a minority one which has no stable and institutionally recognized pedagogical role in the local linguistic community.

Practising the words of languages like Breton in a Speech Technology induced learning environment might be the only available opportunity to learn.

Generally speaking, the sounds of a language are organized in a phonological system which is usually induced by the mother tongue speaker when exposed to the language. The same can hardly happen in second language learning situations: TTS in a lexicon application may help a lot to reduce the gap with L1 speakers basically because the sounds of the language are all spoken in context and what's more in the only legal contexts allowed: the lexicon of that language.

5.1 Using TTS to simulate Interlanguages

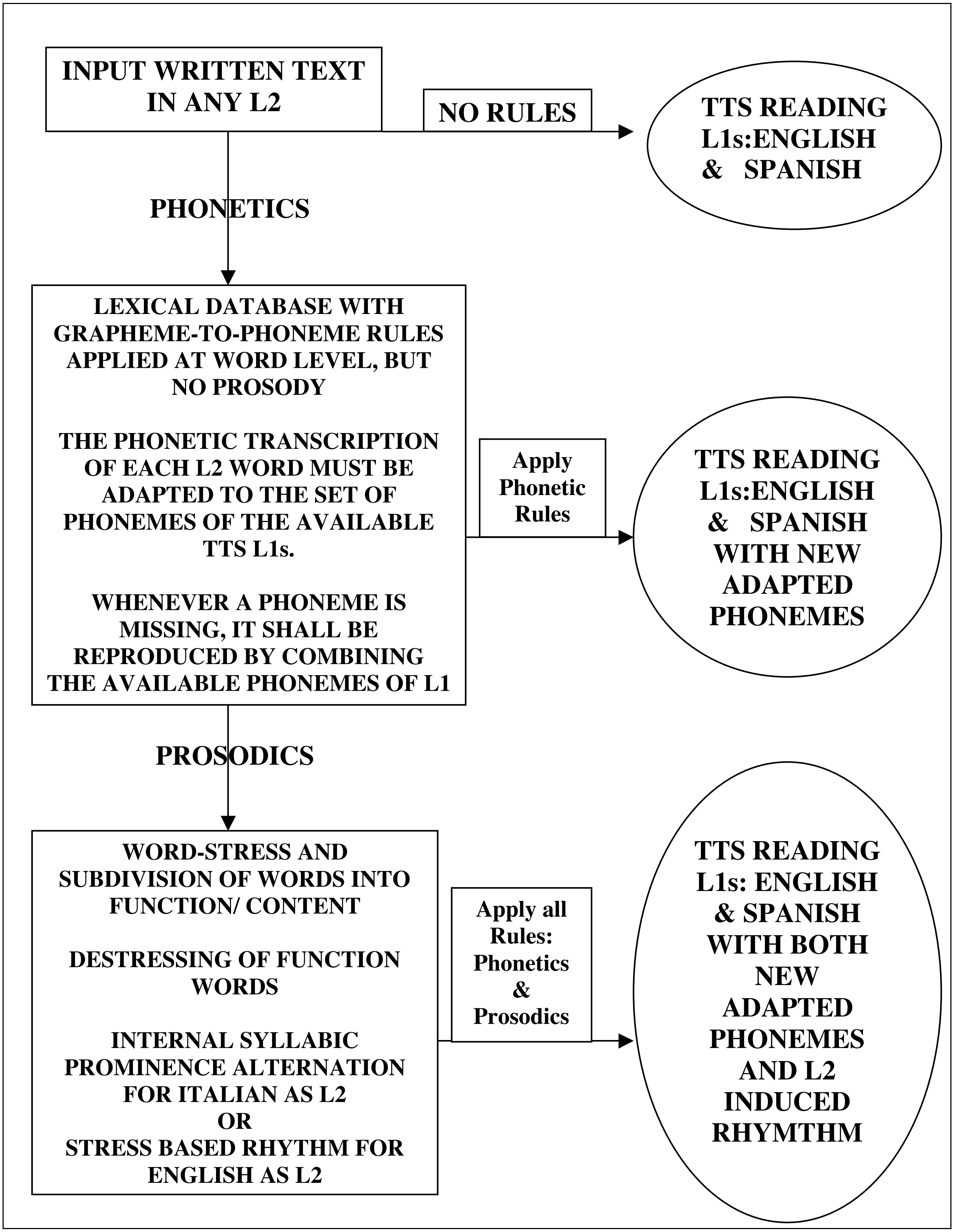
Lexical learning is in our opinion, one of the most interesting application of TTS: besides the mere fact of introducing the student to the pronunciation of sounds in the actual contexts of use, it has the possibility to simulate various stages of Interlanguage. Assuming that when learning a second/foreign language the speaker will at first try to adapt the phonological system he/she already masters - specifically his mother tongue, L1, but also any other system he/she may already have fully learnt. Interlanguage stages will then vary from a Full Beginner's level in which no phonological rule of L2 is used to execute grapheme-to-phoneme translation in a pronunciation task; onto False Beginners, when such rules are known but not all the phonemes of the target system are mastered; to get to an Intermediate Level in which phonemes of L2 are mastered but not all prosodic rules are mastered.

Finally the Advanced quasi-native speaker level with full Phonetic and Prosodic knowledge applied and the ability to differentiate homographs not homophones.

To reach this aim, we have then explored the possibility to modify the interaction of a TTS system by introducing rules of phonetic and prosodic realization which will mimick the interlanguage of a given L1 speaker trying to learn that language as a L2. We worked on such an application by using the Spoken Italian Word List (SIWL) which can be applied to a TTS for English American, and the SLIM Database of English with the TTS available for Spanish American. The idea was to use English American TTS to mimick the interlanguage stages of an American speaker learning Italian from Beginner to Advanced; and to use Spanish American TTS to mimick a Spanish/Romance speaker learning English.

Any TTS application may be fed with pure orthographic input files or with a complete phonetically transcribed version of the orthographic file. In the latter case, grapheme-to-phoneme rules as well as prosodic rhythmic rules may have to be applied first in order to generate the adequate format for the synthesizer.

Here below we show a diagram where the flow of information for a typical TTS system is presented. As appears from the various boxes, in order to provide sufficient information to interact directly with the synthesizer, phonetic and prosodic knowledge is required. To this aim, the Italian words have been converted from graphemes to phonemes and the position of word-stress has been expressedly marked in the input string to allow for the appropriate internal prosody to apply. This has been done by means of a computer program implemented in the beginning of the '80s to serve for the TTS of Italian [8,9,10].


INPUT WRITTEN TEXT IN ANY L2
NO RULES
TTS READING L1s:ENGLISH & SPANISH
PHONETICS
LEXICAL DATABASE WITH GRAPHEME-TO-PHONEME RULES APPLIED AT WORD LEVEL, BUT NO PROSODY
THE PHONETIC TRANSCRIPTION OF EACH L2 WORD MUST BE ADAPTED TO THE SET OF PHONEMES OF THE AVAILABLE TTS L1s.
WHENEVER A PHONEME IS MISSING, IT SHALL BE REPRODUCED BY COMBINING THE AVAILABLE PHONEMES OF L1
Apply Phonetic Rules
TTS READING L1s:ENGLISH & SPANISH WITH NEW ADAPTED PHONEMES
PROSODICS
WORD-STRESS AND SUBDIVISION OF WORDS INTO FUNCTION/ CONTENT
DESTRESSING OF FUNCTION WORDS
INTERNAL SYLLABIC PROMINENCE ALTERNATION FOR ITALIAN AS L2 OR STRESS BASED RHYTHM FOR ENGLISH AS L2
Apply all Rules: Phonetics & Prosodics
TTS READING L1s: ENGLISH & SPANISH WITH BOTH NEW ADAPTED PHONEMES AND L2 INDUCED RHYMTHM

Tab. 5 TTS Interaction with Linguistic Rules for L2 to induce Interlanguage Effects

The effect one gets on first hearing the TTS read the input text, is the same one would get when hearing respectively an English American or a Spanish American read aloud the word list on first glance, i.e. without any previous "training" on problematic words, and with a level of interlanguage competence of Italian comparable to the level of False Beginners.

In particular, all phonemes coinciding with the corresponding phonemes in L1 will be pronounced correctly. But whenever a phoneme is lacking from L1 inventory or simply whenever a phonological rule is not present in the phonological system of L1 the grapheme-to-phoneme rules system available will convert a given set of grapheme into a wrong phonetic equivalent which will cause the hearer to perceive a greater or smaller deviation from the expected pronunciation according to the type of error involved.

When Phonetic Rules are applied, grapheme are turned to the right phoneme, and all phonemes are simulated or reproduced in their almost L1 manner: so this is still understood as a kind of interlanguage where the level of competence for the given L2 has reached an intermediate level, but the prosody is still not mastered. You will get all typical errors at Word-Stress level where stress is placed on the most predictable position - penultimate syllable also when it should be assigned elsewhere. Then, depending on the L1, one will notice typical influences related to the need to apply vowel reductions on unstressed syllables - in case the target language is Italian and the TTS L1 speaking is English - and or the opposite strategy for English as target language L2 and the TTS L1 speaking is Spanish.

As a final simulation, all phonetic and prosodic rules are applied, with the result of inducing the appropriate L2 rhythm and the simulated effect is in some cases totally successful: one hears an absolutely native-like pronunciation of an Advanced student of Italian and English from the TTS of respectively English and Spanish.

The Spoken Italian Word List is made up of 30,000 different words or types with their phonematic and prosodic transcription, lemmata and morpho-syntactic information. The latter set of annotations are very important to define the associated meaning of homographs non homophones. In Italian, we found about 3,000 such homographs some of which receive three different pronunciation patterns. Seen that some words may belong to different categories and to different lemmata, the total amount of pronunciations and meanings available in the SIWL is 48,000.

The following is the list of symbols and corresponding Italian phonemes used in the word list:

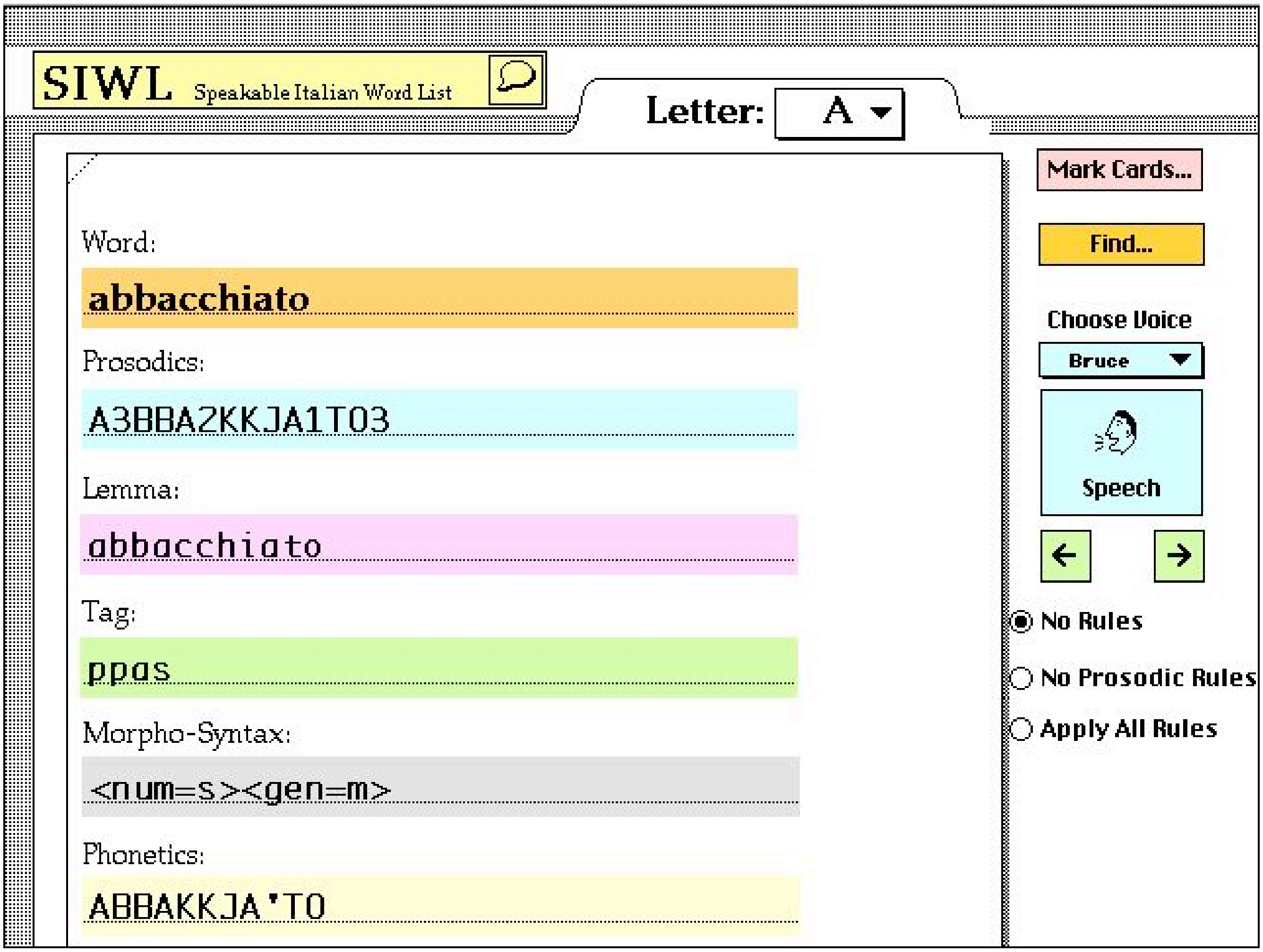
Tab. 6 Grapheme-to-phoneme transcription and their corresponding Italian phonemes

B --> /b/	**A --> /a/**
C --> /tʃ/	**E --> /é/**
K --> /k/	**& --> /è/**
D --> /d/	**O --> /o/**
F --> /f/	**@ --> /ò/**
% --> / Ê/	**I --> /i/**
G --> /g/	**U --> /u/**
< --> / λ/	**> --> / *µ*/**
P --> /p/	**M --> /m/**
S --> /s/	**N --> /n/**
X --> /z/	**R --> /r/**
T --> /t/	**J --> /j/**
V --> /v/	**L --> /l/**

In SIWL application, each word will be pronounced and listened to at three different levels of simulated interlanguage:
aa. Level 1: NO RULES
bb. Level 2: NO PROSODIC RULES
cc. Level 3: APPLY ALL RULES

In Level 1 the synthesized voice uses its own grapheme-to-phoneme rules and associated set of phonemes to produce a reading of the input word. This will result most of the times in a non comprehensible reading due to the distance in the two phonological linguistic systems, the Italian and the English one. From a pedagogical point of view, this fact is very suggestive of the need to study phonetics and prosodics of the L2 the student intends to learn.

In Level 2 the TTS is activated by an internal C program that filters the input word and assigns it a grapheme-to-phoneme transcription with the appropriate internal phonetic symbols used by the Speech Manager. In this way, the synthesized voice will be endowed with the closest possible approximation to the phonetic system of Italian. The reading is now fully intelligible and in some cases also prosodically close to the Italian realization of the input word.



Letter: A

Word: abbacchiato

Prosodics: A3BBA2KKJA1TO3

Lemma: abbacchiato

Tag: ppas

Morpho-Syntax: <num=s><gen=m>

Phonetics: ABBAKKJA'TO

Choose Voice: Bruce

● No Rules
○ No Prosodic Rules
○ Apply All Rules

Finally, in Level 3, all rules are applied, both phonetic and prosodic ones. The latter ones add an Italian-like syllable-based rhythm to the phonetic reading which markedly modifies the quality of the

output. The auditory impression one gets corresponds to an English-American speaker reading Italian with a marked accent - mainly detectable from vowel quality.
In order to produce a prosodically viable rhythm each syllable is marked by a duration and amplitude index which tries to capture the alternation of stressed/unstressed syllables at intraword level. We used 9 different markers which also allow us to differentiate different contexts and interword phenomena. We report here below the list of prosodic markers used in SIWL.

Tab. 7 Prosodic markers introduced in the phonematic transcription of Italian words associated to vowels

1 --> primary stress in open syllable or in syllable closed by /r/
2 --> secondary stress in open syllable
3 --> unstressed open or final syllable
4 --> semivowel
5 --> unstressed syllable in postonic position may also alternante with two unstressed syllables
6 --> primary stress in syllable closed by sonorants - /r/ excluded
7 --> primary stress in closed syllable and in truncated words
8 --> secondary stress in closed syllable
9 --> unstressed closed syllable

The input word representation for the Fitering program is then a mixture of phonematic symbols corresponding to the Italian phonemes with each vowel followed by a prosodic marker:

rivendicano	**RI3V&6NDI3KA5NO3**
roseto	**RO3XE1TO3**
rovescio	**RO3V&7/O3**
seicento	**SE4I3C&6NTO3**
tuoi	**TWO1I8**

SIWL has a FIND function which allows the used to type in a word and get it shown on the screen with the related linguistic information. Alternatively, the user may simply move to and fro from the current word, by pushing on one of the arrows on the right side of the window.

REFERENCES

[1] Delmonte R.(1981), An Automatic Unrestricted Tex-to-Speech Prosodic Translator, Atti del Convegno Annuale A.I.C.A., Pavia, 1075-83.

[2] Delmonte R.(1981), Automatic Word-Stress Patterns Assignment by Rules: a Computer Program for Standard Italian, Proc. IV F.A.S.E. Symposium, 1, ESA, Roma, 153-156.

[3] Delmonte R.(1983), A Phonological Processor for Italian, Proceedings of the 2nd Conference of the European Chapter of ACL,Pisa, 26-34.

[4] Delmonte R.(1983), Elaboratori e linguistica, Lingue e Civiltà, 3; 1(1984), Cladil, Brescia.

[5] Delmonte R.(1984), L'elaboratore nell'insegnamento dell'inglese scientifico, Atti A.I.C.A.:Il calcolatore e la didattica della chimica, AICA,Napoli,67-104.

[6] Delmonte R.,G.A.Mian,G.Tisato(1984), A Text-to-Speech System for the Synthesis of Italian,Proceedings of ICASSP'84, San Diego(Cal), 291-294.

[7] Delmonte R.(1984), On Certain Differences between English and Italian in Phonological Processing and Syntactic Processing, ms., Università di Trieste.

[8] Delmonte R.(1985), Parsing Difficulties & Phonological Processing in Italian, Proceedings of the 2nd Conference of the European Chapter of ACL, Geneva, 136-145.

[9] Delmonte R.G.A.Mian,G.Tisato(1986), A Grammatical Component for a Text-to-Speech System, Proceedings of the ICASSP'86, IEEE, Tokyo,2407-2410.

[10] Delmonte R.(1986), A Computational Model for a text-to-speech translator in Italian, Revue - Informatique et Statistique dans les Sciences humaines, XXII, 1-4, 23-65.

[11] Delmonte R.(1988), Analisi Automatica delle Strutture Prosodiche, in Delmonte R.,Ferrari G., Prodanoff I.(a cura di), Studi di Linguistica Computazionale, Cap.IV, Unipress, Padova, 109-162.

[12] **Delmonte R.**(1990), Semantic Parsing with an LFG-based Lexicon and Conceptual Representations, *Computers & the Humanities*, 5-6, 461-488.

[13] R.Delmonte, R.Dolci(1991), Computing linguistic knowledge for text-to-speech systems with PROSO, **Proc.EUROSPEECH'91**, Genova, 1291-1294.

[14] Delmonte R.(1991), Linguistic Tools for Speech Understanding and Recognition, in P.Laface,R.De Mori(eds), Speech Recognition and Understanding: Recent Advances, Berlin, NATO ASI Series, Vol.F 75, Springer -Verlag, 481-485.

[15] Delmonte R.(1992), Relazioni linguistiche tra la struttura intonativa e quella sintattica e semantica, in E.Cresti et al. Atti del Convegno Internazionale di Studi "Storia e Teoria dell'Interpunzione", Roma, Bulzoni, pp. 409-441.

[16] **Delmonte R., D.Bianchi, E.Pianta,** (1992), GETA_RUN - A General Text Analyzer with Reference Understanding, in *Proc. 3rd Conference on Applied Natural Language Processing - ACL, Systems Demonstrations*, Trento, 9-10.

[17] Delmonte R., F.Greselin(1995), How to create SLIM courseware, in Yeow Chin Yong & Chee Kit Looi(eds.),Proceedings of ICCE '95, Singapore,Applications Track, 206-213.

[18] Delmonte R.(ed.)(1995), How to create SLIM courseware - Software Linguistico Interattivo Multimediale, Unipress, Padova.

[19] Delmonte R. F. Stiffoni, (1995), SIWL - Il Database Parlato della lingua Italiana, Convegno AIA - Gruppo di Fonetica Sperimentale, Trento, 99-116.

[20] Delmonte R., Dan Cristea, Mirela Petrea, Ciprian Bacalu, (1995), PROSODICS - a tool to improve pronunciation of a foreign language, Technical Report, Laboratorio di Linguistica Computazionale, Università di Venezia.

[21] Delmonte R., Dan Cristea, Mirela Petrea, Ciprian Bacalu, Francesco Stiffoni, Modelli Fonetici e Prosodici per SLIM, Atti 6° Convegno GFS-AIA, Roma, 47-58.

[22] Delmonte R., Andrea Cacco, Luisella Romeo, Monica Dan, Max Mangilli-Climpson, Francesco Stiffoni, SLIM - a Model for AUutomatic Tutoring of Language Skills, Ed-Media 96, AACE, Boston.

[23] Delmonte R.(1997), Learning Languages with a "SLIM" Automatic Tutor, in Asiatica Venetiana 2, pp.31-52.

[24] Delmonte R., M.Petrea, C.Bacalu(1977), SLIM Prosodic Module for Learning Activities in a Foreign Language, Proc.ESCA, Eurospeech97, Rhodes, Vol.2, pp.669-672.

[25] Delmonte R.(1998), Prosodic Modeling for Automatic Language Tutors, Proc.STiLL 98, ESCA, Sweden, 57-60.

[26] Delmonte R. (1998), Phonetic and Prosodic Activities in SLIM, an Automatic Language Tutor, Proc.EUROCALL, Leuven,77-78.

[27] Delmonte R. (1998), L'apprendimento delle regole fonologiche inglesi per studenti italiani, in Atti 8° Convegno GFS-AIA, Pisa, 177-191.

[28] **Delmonte R., D.Bianchi** (1998), Dialogues From Texts: How to Generate Answers from a Discourse Model*,* **Atti Convegno Nazionale AI*IA***,* Padova, 139-143.

[29] Bacalu C., Delmonte R. (1999), Prosodic Modeling for Syllable Structures from the VESD - Venice English Syllable Database, in Atti 9° Convegno GFS-AIA, Venezia.

[30] Delmonte R. (1999), A Prosodic Module for Self-Learning Activities, Proc.MATISSE, London, 129-132.

[31] Delmonte R. (1999), La variabilità prosodica: dalla sillaba al contenuto informativo, in Atti 9° Convegno GFS-AIA, Venezia.

[32] Bacalu C., R.Delmonte (1999), Prosodic Modeling for Speech Recognition, in Atti del Workshop AI*IA - "Elaborazione del Linguaggio e Riconoscimento del Parlato", IRST Trento, pp.45-55.

[33] **Bianchi D., R. Delmonte**(1999), Reasoning with A Discourse Model and Conceptual Representations*, Proc. VEXTAL, Unipress, Padova*, 401-411.

[34] Paul Bagshaw, "*Automatic Prosodic Analysis for Computer Aided Pronunciation Teaching*", Unpublished PhD Dissertation, Univ. of Edinburgh, UK, 1994.

[35] Domokos Vékás, Piermarco Bertinetto, "*Controllo vs. compensazione: sui due tipi di isocronia*", in E.Magno Caldognetto e P.Benincà(a cura di*), L'interfaccia tra fonologia e fonetica*, Padova, 1991.

[36] Pier Marco Bertinetto*, Strutture prosodiche dell'italiano*, Accademia della Crusca, Firenze, 1981.

[37] P.C.Bagshaw, S.M.Hiller, M.A.Jack (1993), *"Enhanced pitch tracking and the processing of F0 contours for computer aided intonation teaching",* Proc.Eurospeech93, 1003-1006, Berlin,

[38] Pier Marco Bertinetto, "*The Perception of Stress by Italian Speakers*", Journal of Phonetics, 8, 1980, 385-395.

[39] I.Lehiste(1977), *Isochrony reconsidered*, in Journal of Phonetics 3:253-263.

[40] S.Hiller, E.Rooney, J.Laver and M.Jack(1993), *SPELL: An automated system for computer-aided pronunciation teaching*, Speech Communication, 13:463-473.

[41] Y.Kim, H.Franco, L.Neumeyer(1997), *Automatic Pronunciation Scoring of Specific Phone Segments for Language Instruction*, in Proc. Eurospeech97, Vol.2, 645-648.

[42] D.Klatt(1987), *Review of text-to-speech conversion for English*, J.A.S.A. 82, 737-797.

[43] J.van Santen(1997), *Prosodic Modeling in Text-to-Speech Synthesis*, in Proc. Eurospeech97, Vol.1, 19-28.

[44] J.van Santen, C.Shih, B.Möbius, E.Tzoukermann, M.Tanenblatt, *Multi-lingual durational modeling*, in Proc. Eurospeech97, Vol.5, 2651-2654.

[45] Witt S., S.Young(1998), *Performance Measures for Phone-Level Pronunciation Teaching in CALL*, in Proc. STiLL '98, op.cit., 99-102.

[46] Neumeyer L., F.Horacio, V.Abrash, L.Julia, O.Ronen, H.Bratt, J.Bing, V.Digalakis, M.Rypa (1998), *WebGrader: A Multilingual Pronunciation Practice Tool*, in Proc. STiLL '98, op.cit., 61-64.

[47] Akahane-Yamada R., T.Adachi, H.Kawahara, J.S.Pruitt, E.McDermott(1998), *Toward the Optimization of Computer-based Second language Production Training*, in Proc. STiLL '98, op.cit., 111-114.

[48] Auberg S., N.Correa, V.Locktionova, R.Molitor, M.Rothenberg, (1998), *The Accent Coach: An English Pronunciation Training System for Japanese Speakers*, in Proc. STiLL '98, op.cit., 103-106.

[49] Price P. (1998), *How can Speech Technology Replicate and Complement Good Language Teachers to Help People Learn Language?,* in Proc. STiLL '98, op.cit., 103-106.

[50] Eskenazi M. & S.Hansma(1998), *The Fluency Pronunciation Trainer*, in Proc. STiLL '98, op.cit., 77-80.

[51] Delcloque P., C.Campbell(1998), *An intelligent tutor for the acquisition of French pronunciation within the communicative approach to language learning. The secondary and tertiary solution,* in Proc. STiLL '98, op.cit., 9-12.

[52] Meador J., F.Ehsani, K.Egan, S.Stokowski(1998), *An Interactive Dialog System for Learning Japanese*, in Proc. STiLL '98, op.cit., 65-69.
[53] Auberg S., N.Correa, M.Rothenberg, M.Shanahan(1998), *Vowel and Intonation Training in an English Pronunciation Tutor*, in Proc. STiLL '98, op.cit., 69-73.
[54] Ueyama M.(1997), *The Phonology and Phonetics of Second Language Intonation: The Case of "Japanese English"*, in Proc ESCA'97, Vol.5, 2411-2414.
[55] CALICO Journal(1999), 16, 3, Special Issue - Tutors that Listen: Speech Recognition for Language Learning.
[56] **Herzog O., C.-R. Rollinger**(1991)(eds), **Text Understanding in LILOG**, Springer Verlag, Berlin.
[57] **Jackendoff R.** (1983), **Semantics and Cognition**, MIT Press, Cambridge Mass.
[58] **Jackendoff R.** (1985), Multiple Subcategorization and theTheta-Criterion: the Case of *Climb*, *Natural Language and Linguistic Theory* 3, 271-296.
[59] **Jackendoff R.**(1987), **Consciousness and the Computational Mind**, The MIT Press, Cambridge Mass.
[60] **Jackendoff R.**(1987), The Status of Thematic Relations in Linguistic Theory, *Linguistic Inquiry* 18, 369-411.
[61] **Jackendoff R.**(1993), On the role of Conceptual Structure in Argument Selection: A Reply to Emonds, *Natural Language and Linguistic Theory* 11, 2, 279-312.
[62] **Webber B. L.** (1981), Discourse model synthesis: preliminaries to reference. In A. Joshi, B. L. Webber and I. Sag (eds.), **op.cit.**

APPENDIX
Websites Related to Speech Synthesis

This is a sample list of websites some of which, the first two, point to a number of other websites. We did not intend to produce a complete list, however we assume that most of relevant sites are here.

URL=http://www.itl.atr.co.jp/cocosda/synthesis/links.html

Speech Synthesis Links

- CHATR (ATR's speech synthesis system)
- The Epos Speech Synthesis System
- Infovox
- COST-258
- JEIDA Guidelines for Speech Synthesizer Assessment
- JEIDA IFA Features Q5.3:
- References/Books on Synthesis SpeechLinks:
- Speech Synthesis Museum of Speech Analysis and Synthesis
- ICG Grenoble's `exemples sonores'
- Speech Synthesis at ICP Grenoble
- Microsoft's Speech Synthesis Project
- Speech Synthesis Software/Hardware
- The MBROLA project homepage
- Bibliography Search on Speech Processing
- Speech/Acoustic related WWW information list
- Lector (Spanish)
- AT&T Advanced Speech Products Group
- AT&T's WATSON Demo
- Lucent Technologies
- Bell Labs Text-to-Speech System
- Orator (Bellcore)
- The Birmingham Speech Synthesis Systems museum
- CSLU HLT Survey OGI's tts research
- CSLU Speech Synthesis Research Group
- CSTR Speech Synthesis Links
- EUROVOCS Eloquent Technology, Inc.
- A Speaking Web Site First Byte Text-To-Speech
- HOME PAGE ICP (Bailly)
- Hadifix (Bonn) Stuttgart's Synthesis Collection
- BT Laboratories - Text-to-Speech
- NTT's Japanese synthesis
- Telia AT&T Research Voices (c) 1996 AT&T
- MicrosoftIBM Voicetype
- Apple Speech Home Apple's PlainTalk
- Multimodal Speech Synthesis from KTH
- Speech Synthesis - Speech Toys SoftVoice, Inc. homepage
- WebSpeak (de Pijper) Bibliography Phonetics / Speech Technology Say...

URL=http://fonsg3.let.uva.nl/IFA-Features.html

Speech Synthesis

* Bell-Labs (Lucent Technologies) - Demonstrations from the National Center for
  Voice and Speech
* Special features from Haskins Laboratories
* Haskins Laboratories
* More Speech on the Web from the IPO, the Institute for Perception Research
  with extensive links
* Audio-demonstrations from the OTS Phonetics department
  (at OTS the Research Institute for Language and Speech, Utrecht University)
* YorkTalk, the speech generation system under development at the University of York UK
* Demonstration of Synthetic Vowels from Automatic Speech Recognition
  Lab at the Beckman Institute of the University of Illinois
* Demonstrations from TMH, at KTH, Stockholm, Sweden
* EUROVOCS demonstration in Ghent, Belgium
* Institut de la Communication Parle Grenoble, FRANCE
* Musee sonore de la synthese de la Parole en francais
* Museum of Speech Analysis and Synthesis
* Talking Faces, lips and more
* The Museum of Machine Generated Speech and Singing
* rsynth a public domain speech synthesizer
* Say..., Speech synthesis demonstration from Tios, Informatica, at the University of Twente
* Synth: Speech Examples, a museum in Birmingham
* Travelers' Japanese with Voice

URL=http://www.cstr.ed.ac.uk/~awb/synthesizers.html
URL=http://www.ultranet.com/~rongemma/sites.htm
URL=http://www.dcs.shef.ac.uk/research/groups/spandh/world/misclinks.html
URL=http://www-a2k.is.tokushima-u.ac.jp/member/kita/NLP/index.html
URL=http://www.ims.uni-stuttgart.de/phonetik/synthesis/
URL=http://www.tue.nl/ipo/hearing/webspeak.htm
URL=http://lorien.die.upm.es/research/synthesis/synthesis.html
URL=http://mambo.ucsc.edu/psl/speech.html