

Université du Maine
Département informatique

Travaux présentés devant
l'université du Maine par

Yves Aubry

en vue d'obtenir le
D.R.T. d'ingénierie
mention informatique

**Logiciel de traitement de la parole
et d'aide à l'enseignement
et à l'apprentissage de la prosodie:
application au breton.**

Soutenu le 19 novembre 2004 devant la commission d'examen:

Marc Baudry, professeur à l'université du Maine:	directeur des travaux
Guy Mercier, chercheur associé à l'université de Rennes II:	directeur des travaux
Francis Favereau, professeur à l'université de Rennes II:	rapporteur
Dominique Ruhlmann, enseignant détaché au CRDP de Bretagne:	rapporteur
Yannick Estève, maître de conférence à l'université du Maine:	examineur
Jean Le Clerc de la Herverie, responsable éditorial de TES:	examineur
Wolfgang Hess, professeur à l'institut de phonétique de l'université de Bonn:	examineur

Université du Maine
Département informatique

Travaux présentés devant
l'université du Maine par

Yves Aubry

en vue d'obtenir le
D.R.T. d'ingénierie
mention informatique

**Logiciel de traitement de la parole
et d'aide à l'enseignement
et à l'apprentissage de la prosodie:
application au breton.**

Soutenu le 19 novembre 2004 devant la commission d'examen:

Marc Baudry, professeur à l'université du Maine:	directeur des travaux
Guy Mercier, chercheur associé à l'université de Rennes II:	directeur des travaux
Francis Favereau, professeur à l'université de Rennes II:	rapporteur
Dominique Ruhlmann, enseignant détaché au CRDP de Bretagne:	rapporteur
Yannick Estève, maître de conférence à l'université du Maine:	examineur
Jean Le Clerc de la Herverie, responsable éditorial de TES:	examineur
Wolfgang Hess, professeur à l'institut de phonétique de l'université de Bonn:	examineur

Travail effectué à: TES
30, rue Brizeux
22000 St. Brieuc

Glossaire

D.R.T.	Diplôme de Recherche Technologique.
T.E.S.	Ti-Embann ar Skoliou Brezhonek, centre régional multimédia de production pédagogique en langue bretonne.
Prosodie	La prosodie est représentée par le rythme (mise en relief des syllabes accentuées grâce à la combinaison des paramètres acoustiques de durée, de hauteur et d'amplitude; en français, l'accentuation est marquée par un allongement de la durée de la syllabe accentuée) et par l'intonation.
Phonèmes	Ce sont les sons élémentaires du langage. Leur nombre est variable d'une langue à une autre. On dénombre 37 phonèmes pour le français; pour le breton on peut compter 49 phonèmes auxquels on peut rajouter 15 voyelles longues et 15 diphtongues. Les phonèmes se répartissent en différentes classes. On distingue par exemple les voyelles, qui sont voisées, et les consonnes, qui peuvent être voisées ou sourdes.
Pitch	Fréquence fondamentale. Elle correspond aux oscillations des cordes vocales et est notée f_0 ou F_0 .
Voisé	Se dit d'un son produit par la mise en vibration des cordes vocales. Toutes les voyelles du français sont voisées, ainsi que certaines consonnes.
Spectrogramme	Représentation fréquentielle d'un signal basée sur l'analyse de Fourier.
Cepstre	Méthode d'analyse de la parole permettant de séparer les caractéristiques de l'excitation de celles du conduit vocal; la méthode fait appel à une transformée de Fourier et à une transformée de Fourier inverse. Les coefficients du cepstre sont obtenus dans le domaine temporel (quéfrentiel).
Vocodeur à canaux	Analyse spectrale de la parole à l'aide de bancs de filtres répartis dans la gamme de fréquence de la parole.
Prédiction linéaire	Méthode d'analyse de la parole basée sur le modèle source – conduit vocal; le signal de parole est prédit à l'aide d'une fonction linéaire de signaux précédents.
Sampa	Nom du code phonétique utilisé par le synthétiseur vocal Mbrola.

Snack	Librairie pour Tcl/Tk qui permet de lire, d'enregistrer des fichiers sonores ou encore de les afficher suivant leur représentation temporelle ou spectrale.
Tcl/Tk	Langage de programmation utilisé en particulier pour réaliser des interfaces graphiques.
Visual C++	Environnement de développement d'applications pour Windows, basé sur le langage de programmation C++.
WAV	Waveform, format de fichier audio très utilisé sous Microsoft Windows.

Remerciements

Je tiens à exprimer ma reconnaissance à Marc Baudry mon directeur de D. R. T. à l'université du Mans qui m'a permis de poursuivre ce travail de recherche technologique. Je remercie Guy Mercier pour ses conseils et connaissances apportées dans le domaine de la reconnaissance et du traitement de parole, ainsi que Jean Le Clerc de la Herverie responsable d'édition de TES et Gilles Godefroy directeur du CDDP de St Brieuc pour leur soutien durant le projet.

Je remercie également Pierre Lavanant, Ronan L'Hourre ainsi que toutes les autres personnes qui ont testé le logiciel et m'ont apporté leurs critiques et leurs suggestions pour l'améliorer.

Je remercie aussi les personnes m'ayant précédé sur son développement : Philippe Parnet, Guillaume Mocquart, Aurélien Guillou et Florent Moullet.

Enfin, je remercie le personnel technique et administratif de l'ENSSAT qui m'a accueilli et aidé durant deux ans et demi ainsi que le personnel de TES qui m'a accueilli au début de cette année.

Résumé

Ce projet est mené par la maison d'édition TES¹. TES est une maison de production de matériel pédagogique localisée à St-Brieuc qui fournit aux écoles enseignant le breton, un ensemble de matériels et d'équipements pédagogiques.

T.E.S. participe entre autre au développement d'un système de synthèse vocale du Breton, d'un dictionnaire vocal Français - Breton utilisant cette synthèse vocale, et du correcteur de prosodie.

Le correcteur de prosodie a pour but d'aider les élèves à apprendre la prosodie de la langue bretonne. Cela passe par un apprentissage de l'intonation de la parole, du rythme et de l'accentuation.

Le correcteur visualise et compare ces caractéristiques du signal de parole de l'élève avec celles d'un enregistrement de référence. Ces caractéristiques sont affichées dans des fenêtres séparées. Ce logiciel permettra de détecter et de signaler les erreurs commises.

Le logiciel comprend une partie "*création d'exercices*" destinée à l'enseignant qui peut s'enregistrer et créer des exercices qui serviront de modèles de référence pour l'élève et une partie "*pratique de la prosodie*" destinée à comparer la parole de l'élève avec celle du maître. Ces modules font appel aux techniques d'analyse de la parole, de synthèse et de segmentation en phonèmes et en syllabes.

Pour rendre le logiciel plus convivial, les courbes prosodiques (Pitch, énergie) sont synchronisées, normalisées et peuvent être affichées dans la même fenêtre sur l'écran de l'ordinateur.

Pour le moment ce logiciel est un outil général, il manque encore l'aspect pédagogique lui-même qui est un volet très important pouvant comporter différents ensembles d'exercices en fonction du public et du niveau visé. Ce volet sera mis en place progressivement en concertation avec les enseignants.

¹Ti-Embann ar Skoliou Brezhonek, centre régional multimédia de production pédagogique en langue bretonne.

Abstract :

This project was managed by TES who is an editor for Breton language speaking schools. TES is located in Saint-Brieuc (Brittany) and is providing these schools with teaching books, C.D., K7 and more recently educational software tools.

Among other software tools, TES is working on a text-to-speech synthesis system for Breton, on a bilingual spoken dictionary based on Speech Synthesis and on an Accent Tutor.

The aim of the accent tutor which is described in this report is to help learners to understand what is prosody and to provide them with comprehensible visualisation of all three components of prosody: intonation, stress and rhythm.

Relevant features of the tutor's (respectively learner's) speech signals are visualised in separated windows and compared in order to pinpoint what is mispronounced by the student.

This software is composed of two parts; in the first component named "*building of exercises*", the teacher is asked to speak and to record his utterances which are then segmented and converted into reference exercises. In the second component named "*practice of prosody*", the learner is asked to listen to reference signals and then to record his own utterances which are themselves segmented, compared to the tutor's speech and visualised in a separated window. Speech signal processing techniques, speech synthesis and techniques allowing to segment speech signals into phonemes and syllables are developed in these modules.

In order to make this software more easy to handle, prosodic curves like pitch and energy curves can be synchronised and visualised in the same screen window.

This software is still under development; dedicated exercises fitting to the learner's level have to be added together with algorithms able to detect irrelevant prosodic features and with a module explaining the differences and how to overcome them. This will be put progressively in position with the help of teachers.

Berradur

Lañset eo bet ar mennad-mañ gant T.E.S., Ti Embann ar Skolioù brezhoneg, staliet e Sant-Brieg. E-barzh TES e vez krouet danvezhioù-keleñn evit ar skolioù a zo o keleñn brezhoneg.

Gant sikour Skol Veur Roazhon 2, IRISA Lannuon, ENSSAT ha Skol Vreizh o deus labouret tud TES war sintezenn ar gomz evit ar brezhoneg, war "Ar Geriadur A Gomz" (ar brezhoneg a-vremañ) ha war ul lojisiel evit deskiñ ar brozodiezh : difazier ar brozodiezh.

Sanset eo an difazier-se sikour ar skolidi da gompren petra eo ar brozodiezh (an taol-mouezh, al lusk, ar pouez-mouezh) ha da wellaat o brezhoneg. Diskouez ha keñveriañ (e-barzh daou brenestr) arouez an desker hag arouez ar mestr a ra an difazier. Pa vo echu e vo gouest an desker da verzout an diforc'hoù.

Kavet e vez div lodenn e-barzh al lojisiel. E-barzh ul lodenn anvet "sevel poelladennoù" a c'hall ar mestr en em enrollañ ha sevel poelladennoù prest da servij evel skouerioù evit ar skolidi pe ar studierien. E-barzh ul lodenn all anvet "labourat gant ar brozodiezh" e vez diskouezet ha keñveriet komz an desker ha komz ar mestr. Teknikoù evit dielfennañ ar gomz, evit sintetizañ hag evit troc'hañ al lavar e fonemoù pe e silabennoù a vez kavet e-barzh an div lodenn-se.

Kromennoù ar brozodiezh reolenet ha kenamzeriet a vez gwelet er memes prenestr evit ma vefe aezetoc'h implij al lojisiel-se.

Betek bremañ al lojisiel-se n'eo c'hoazh nemet ur benveg. Mankout a ra ar bedagogiezh a zo a-bouez bras evit sevel poelladennoù disheñvel hervez live yezh an deskidi hag ar pal da dizout. Gant sikour ar gelennerien e vo pleustret tamm ha tamm war ar mankoù-se.

Sommaire

Glossaire.....	1
Remerciements.....	3
Résumé.....	4
Abstract :.....	5
Berradur.....	6
Sommaire.....	7
1. Introduction.....	10
1.1 Situation de la langue bretonne.....	10
1.2 Traitement de la parole.....	10
1.2.1 Logiciels d'apprentissage de langue	11
1.2.2 Prosodie.....	11
1.2.3 Lancement du projet.....	12
2. Correcteur de prosodie.....	16
2.1 État des travaux.....	16
2.1.1 Exercices.....	17
2.1.2 Création manuelle d'exercices.....	18
2.1.3 Création automatique d'exercices de synthèse.....	19
2.1.4 Pratique de la prosodie.....	21
2.2 Choix techniques.....	22
2.2.1 Interfaçage entre Tcl/Tk et C.....	23
3. Comparaison et évaluation des signaux.....	25
3.1 Architecture générale.....	25
3.2 Calcul des caractéristiques prosodiques.....	25
3.2.1 Calcul du pitch et de l'énergie.....	26
3.2.1 Calcul du cepstre.....	27
3.3 Alignement automatique.....	29
3.3.1 Distances locales.....	30
3.3.2 Distances cumulées.....	30
3.3.3 Chemin optimal.....	32
3.3.4 Détection de début et fin de parole.....	33
3.3.5 Résultats obtenus:.....	34
3.4 Interface de pratique de la prosodie.....	34
3.4.1 Affichage des courbes sonores.....	35

3.4.2	Sélection simultanée de plages du signal de parole.....	36
3.4.3	Affichage du pitch ou de l'énergie de l'élève dans la zone maître.....	37
3.4.4	Affichage par phonème, syllabe ou mot.....	37
3.4.5	Indication des erreurs d'intonation.....	38
3.4.6	Autres fonctionnalités.....	39
4.	Synthèse vocale.....	40
4.1	Principe.....	40
4.2	Prétraitements.....	41
4.2.1	Traitement des nombres.....	42
4.3	Transcription graphèmes - phonèmes.....	42
4.4	Génération de la prosodie.....	43
4.4.1	Principe.....	43
4.4.2	Analyse grammaticale.....	44
4.5	Synthèse acoustique.....	45
4.5.1	La méthode TD-Psola.....	46
4.5.2	Caractéristiques principales de la méthode MBROLA.....	46
4.5.3	Synthèse par unités variables.....	46
4.6	Interface du système de synthèse.....	48
4.7	Fichiers de sortie du système de synthèse vocale.....	49
4.8	Imitation de la parole d'un locuteur par la synthèse vocale.....	51
5.	Création d'exercices.....	53
5.1	Principe.....	53
5.2	Création automatique d'exercices de synthèse.....	54
5.2.1	Interface de création automatique.....	54
5.2.2	Intégration de la synthèse dans le correcteur.....	55
5.2.3	Création des exercices de synthèse.....	55
5.3	Comparaison maître-synthèse.....	56
5.3.1	Modification des frontières.....	57
5.4	Module de modification des frontières d'un exercice.....	58
5.4.1	Utilisation.....	58
5.4.2	Programmation.....	59
6.	Réalisation d'un premier cdrom de test.....	60
6.1	Procédure d'installation.....	60
6.1.1	Organisation des répertoires.....	60
6.1.2	Les différentes étapes de l'installation.....	60
6.2	Aide en ligne.....	61
6.2.1	Technique.....	61
6.2.2	L'aide du correcteur de prosodie.....	63
7.	Conclusion et évolution.....	64

8. Bibliographie.....	67
ANNEXES.....	75
Exemple de fichier d'exercices.....	76
Présentation brève du langage Tcl/Tk	78
Le code Sampa.....	81
Règles de transcription en phonétique.....	82

1. Introduction

1.1 Situation de la langue bretonne

Au début du vingtième siècle, 75% de la population bretonne située dans la moitié ouest de la Bretagne (Basse-Bretagne) était composée de bretonnants. En 1990, ce pourcentage était tombé à 17%. Aujourd'hui, sur les 268000 personnes dont le breton est la langue maternelle, 2000 seulement ont moins de 30 ans.

D'un autre côté, l'enseignement du breton progresse et, en moins de 20 ans, les effectifs des classes bilingues sont passés de quelques centaines à 8850 élèves et le nombre d'élèves qui bénéficient de cours de breton à l'école s'élève à 23000.

La progression de l'apprentissage du breton se heurte pourtant à une difficulté majeure : l'absence d'environnement linguistique. Il est possible de vivre en Basse-Bretagne sans entendre parler breton; la plupart des parents d'élèves apprenant le breton ne sont pas bretonnants de naissance et il en va de même pour les enseignants. Dans un tel contexte, les élèves et parfois les enseignants commettent de graves erreurs de syntaxe et de prononciation.

1.2 Traitement de la parole

Au milieu des années 1990, la technologie de la parole est sortie des laboratoires. Les logiciels de reconnaissance de la parole sont devenus plus fiables et plus faciles à utiliser. Actuellement la plupart des systèmes de reconnaissance sont indépendants du locuteur. Les unités de base de la reconnaissance sont des unités phonétiques contextuelles (allophones) et elles sont modélisées par des modèles de Markov cachés (Hidden Markov Models, H.M.M. [Rabiner, 1994]) dont les paramètres sont estimés pendant la phase d'apprentissage, à partir d'un corpus de parole étiqueté, de grande taille. L'ajout d'informations linguistiques (syntaxiques, sémantiques, pragmatiques) permet de réduire le pourcentage d'erreurs de reconnaissance de mots. Ces systèmes sont en général mis au point pour une langue donnée. Leur utilisation pour une autre langue demande un investissement important comme par exemple la collecte et l'étiquetage de bases de données de parole comportant des ensembles de mots et de phrases prononcés par un large échantillon de locuteurs; ce travail a été réalisé pour les langues dominantes mais pour les langues minoritaires moins intéressantes commercialement, il reste encore un gros effort à fournir. Un autre inconvénient pour l'utilisation de la reconnaissance dans les logiciels d'apprentissage de langues tient au fait que les algorithmes de reconnaissance ne cherchent pas à évaluer la qualité de la prononciation du locuteur en acceptant les prononciations correctes et en rejetant les prononciations incorrectes ; au contraire, plus ils sont capables de reconnaître des

phrases prononcées de manière différente, plus ils sont performants. Il faut donc adapter ces logiciels pour la tâche d'apprentissage d'une langue.

Dans le même temps, la nouvelle technique de synthèse par concaténation d'unités de taille variable [Hess, 2000] remplaçant la technique de synthèse par concaténation de diphtonges a permis d'améliorer de façon significative la qualité de la synthèse de la parole.

1.2.1 Logiciels d'apprentissage de langue

Dans les années 80-90, des logiciels de langue ayant recours à l'affichage de courbes graphiques pour aider l'apprenant à établir un lien entre sa production orale et celle proposée comme référence ont commencé à sortir des laboratoires (Wincécil, Winpitch, Snorri [Laprie, 1999], etc.).

Maintenant d'autres logiciels vont plus loin et intègrent les techniques de reconnaissance de la parole pour évaluer la prononciation de l'apprenant et détecter les erreurs. Cependant les logiciels du commerce présentent encore de nombreuses lacunes : limites technologiques, modes d'utilisation et affichages compliqués, peu adaptés aux enseignants ou aux élèves, méthodes de diagnostic, de notation, d'explication inadéquates, contenu des leçons ou des exercices, limité, fermé, pas assez évolutif. L'interaction dans les logiciels du commerce est assez souvent rudimentaire.

1.2.2 Prosodie

Les recherches en linguistique ont montré que l'intonation et les caractéristiques prosodiques¹ sont des composantes indispensables de la langue et de la fonction de communication [P. Martin, 1987]. Les paramètres prosodiques de la langue maternelle et en particulier l'intonation sont perçus et reproduits très tôt par l'enfant ([CHUN, 1998]; [KONOPZYNSKI, 1999]). L'enfant éprouve peu de difficultés pour acquérir les caractéristiques prosodiques d'une deuxième langue. Pour les élèves plus âgés et pour les adultes, par contre, l'apprentissage de la prosodie, c'est-à-dire de l'intonation et du rythme d'une deuxième langue est beaucoup plus difficile et nécessite une rééducation prosodique. Ces caractéristiques ont été jusqu'ici trop largement sous-estimées dans l'enseignement des langues. De plus, la prosodie est souvent considérée comme secondaire, alors qu'en réalité, elle joue un rôle fondamental dans l'apprentissage de la prononciation d'une langue: elle constitue en effet une véritable "structure d'accueil" ([KONOPCZYNSKI, 1999]), à l'intérieur de laquelle le système phonologique va pouvoir s'organiser. Si la prosodie n'est pas correctement mise en place dès le début de l'apprentissage, la prononciation des voyelles et des consonnes risque d'en être gravement affectée. Loin d'être un phénomène accessoire, elle est une étape fondamentale dans l'apprentissage d'une langue étrangère.

¹ Voir glossaire page 1

1.2.3 Lancement du projet

En 1994, le responsable des collections à T.E.S. (Ti Embann ar Skolioù Brezhonek, maison d'édition pour les écoles bretonnes), R. Le Coadic, des personnes de Skol Vreizh, des enseignants linguistes (université de Rennes II et de Lampeter au pays de Galles), des chercheurs de l'I.R.I.S.A. (Institut de recherches en Informatique et Systèmes Aléatoires), enseignants à l'ENSSAT et des ingénieurs d'Alcatel et du C.N.E.T. (France Télécom), ayant travaillé dans le domaine du traitement du signal, de la parole ou dans l'informatique s'unissent pour former un groupe de travail informel. Ce groupe de travail se réunissant tous les mois essaie de faire le point sur l'état des différentes technologies de l'époque et sur les besoins pédagogiques des enseignants et des élèves des différentes filières de l'éducation (Diwan, écoles bilingues du public et du privé) enseignant le breton et en breton.

En 1995, T.E.S., l'I.R.I.S.A., l'université de Rennes II, Skol Vreizh et les membres du groupe de travail décident de coopérer pour développer de nouveaux outils pédagogiques intégrant les technologies de l'information et les technologies vocales afin de tirer le meilleur parti de la rapidité, de la fiabilité et de la robustesse des techniques de traitement de la parole (analyse, synthèse, reconnaissance, visualisation) et pour lancer le projet K.G.B. (Kenaos ar Gomz e Brezhoneg, synthèse de la parole en breton) dans le cadre du projet CORDIAL de l'IRISA.

Mais les moyens humains, financiers et techniques sont bien plus limités pour les langues minoritaires que pour les langues nationales et il en va de même pour les ressources techniques et linguistiques, l'équipe décide donc de procéder par étapes avec des objectifs limités mais réalistes pouvant aboutir à des produits finis et utilisables dans un temps raisonnable.

Pour commencer, le projet a pu bénéficier d'un premier travail exploratoire sur la transcription graphèmes phonèmes du breton réalisé quelques années auparavant par H. Le Borgne et L. Le Guillouzer ainsi que des travaux de M. Guyomard et de M. Divay sur la transcription graphèmes – phonèmes pour la langue française [**DIVAY, GUYOMARD, 1972**]. Quelques essais de synthèse des nombres bretons et de synthèse d'informations météo avaient également été réalisés au CNET en 1990 par C. Hamon. Mais le travail le plus important dont a pu bénéficier le projet a été le logiciel de transcription graphèmes - phonèmes réalisé par P. Lintanf, en 1994 [**LINTANF, 1994**] sous la direction de F. Favereau (université de Rennes II) et de M. Divay (I.U.T., Lannion). Ce travail est repris en 1995 dans le projet CORDIAL de l'IRISA à Lannion, par Jean-Luc Tromparent [**Tromparent, 1995**], stagiaire de DEA. En 1996, dans le cadre d'un stage à l'université de Limerick, H. Gourmelon réécrit la technique de synthèse TD/PSOLA de concaténation de diphtongues et de modification des paramètres prosodiques [**HAMON, 1989**], en langage Visual C++. Grâce à cet outil, H. Gourmelon peut synthétiser quelques phrases en breton [**GOURMELON, 1996**].

En même temps, comme l'université de Mons offrait des conditions intéressantes, tant au point de vue technique que financier pour utiliser la technique de synthèse de concaténation de diphtongues *MBROLA* (Dutoit, 1997), les partenaires du projet KGB décident de privilégier cette technique. Un premier corpus de logatomes (mots sans signification), de mots et de phrases usuels contenant l'ensemble des diphtongues de la langue bretonne est élaboré à partir de l'inventaire des différents phonèmes du breton. A. Ebrel enregistre le corpus dans les studios Louis Carsin à Saint-Brieuc. R. Sokol,

stagiaire dans le groupe **CORDIAL** procède à la numérisation du corpus [Sokol, 1996] sur les ordinateurs de l'**ENSSAT** et à sa segmentation en phonèmes et en diphtongues à l'aide du logiciel de segmentation de la parole **SNORRI**, mis au point par Y. Laprie au **CRIN** à l'université de Nancy [Laprie, 1999]. Grâce à ce logiciel, les marques de pitch nécessaires à la technique **PSOLA** y sont également insérées. Le nouveau corpus segmenté est mis en forme et envoyé à l'université de Mons qui lui fait subir un ensemble de traitements (mise à pitch constant, etc.) pour qu'il soit utilisable par le logiciel **MBROLA**. Ainsi dès 1997, l'équipe du projet KGB dispose d'une base de diphtongues et peut envisager la possibilité de réaliser la synthèse de n'importe quelle phrase bretonne ; il reste à adapter la transcription graphèmes – phonèmes du breton à la technique **MBROLA** (passage du format IPA au format **SAMPA** utilisé par **MBROLA** et inclusion des marques de prosodie : durée des phonèmes et valeur de hauteur des phonèmes voisés), à obtenir un bon modèle prosodique de la phrase bretonne et à écrire le logiciel décrivant ce modèle.

D'autre part, TES souhaitait disposer rapidement d'un premier logiciel pédagogique, utilisable dans les écoles. Dans cette optique, Skol Vreizh par l'intermédiaire de P. Lavanant et de F. Favereau met à la disposition des chercheurs les fichiers du dictionnaire bilingue de F. Favereau (*Geriadur ar Brezhoneg a vremañ*, dictionnaire du breton contemporain, [Favereau, 1993]), au format *Word 2*. H. Gourmelon et J. P. Messenger [Gourmelon, 1999] transcrivent ce dictionnaire au format *R.T.F.* (Rich Text Format) et un gros travail impliquant plusieurs personnes bénévoles et quelques stagiaires est réalisé pour détecter les différents champs (traduction, définitions, exemples d'usage, variantes phonétiques et orthographiques, abréviations) et pour restructurer les définitions. J.P. Messenger met au point une première version Web du dictionnaire qui depuis a été reprise en Californie pour en faire une version internet.

Une interface conviviale de consultation du dictionnaire pour Windows, écrite en *Visual C++* est conçue et développée par X. Madigou [Madigou, 1997] étudiant stagiaire de l'**ENSSAT**; H. Gourmelon écrit un premier modèle prosodique pour la prononciation des mots et intègre dans le dictionnaire le moteur de synthèse **MBROLA** et la base de diphtongues pour procéder à la synthèse des mots du dictionnaire à partir de leur transcription phonétique. La version V1 de ce dictionnaire et du logiciel de consultation sort en décembre 98, sous la forme d'un CDROM. Ce dictionnaire a été distribué dans les écoles et commercialisé par Skol Vreizh. C'est le premier logiciel éducatif réalisé par l'équipe du projet KGB.

Cependant, ce logiciel n'était pas un logiciel destiné à l'enseignement ou à l'apprentissage de la prosodie. En 1996, J. P. Messenger publie une étude bibliographique sur l'utilisation des marques prosodiques utilisées en synthèse de la parole comme moyen de faire ressortir l'influence des paramètres prosodiques sur la qualité de la synthèse. Après le départ de J. P. Messenger, c'est une direction légèrement différente qui est prise en 1997 avec P. Parnet, étudiant stagiaire. L'objectif est d'écrire un logiciel permettant de comparer un signal de parole "modèle" (préalablement enregistré ou prononcé par l'enseignant) au signal de parole de l'apprenant dans le but de détecter automatiquement les différences et de trouver les écarts de prononciation, avec l'aide d'un enseignant. Cette première version est constituée de programmes de traitement et de visualisation du signal (FFT, calcul des coefficients du cepstre, calcul de l'énergie et du pitch) et d'une première interface simplifiée pour l'affichage des signaux [Parnet, 1998]. Ce travail sera poursuivi les années suivantes par de nouveaux stagiaires, G.

Mocquard [**Mocquard, 1999**], A. Guillou [**Guillou, 2000**] et F. Moullet [**Moullet, 2001**] et repris pour donner la version actuelle du correcteur de prosodie décrite dans ce rapport.

En parallèle, le travail sur la synthèse à partir du texte se poursuit. Lors de mes divers stages dans l'équipe, j'écris une nouvelle interface en langage Delphi, j'améliore la transcription graphèmes-phonèmes et intègre un premier modèle prosodique simplifié pour la synthèse des phrases, réalisant ainsi la première version du logiciel de synthèse en breton à partir du texte. Même si le modèle prosodique est encore rudimentaire, ce logiciel présente un certain nombre d'avantages et en particulier une interface conviviale permettant à l'utilisateur de modifier les règles de transcription phonétique ou d'en ajouter de nouvelles; il peut aussi manipuler et modifier les paramètres prosodiques, introduire ses propres pauses ou supprimer certains phonèmes. Ce logiciel peut également synthétiser les nombres et dispose d'une première analyse grammaticale. Il faut encore lui ajouter une analyse morphologique puis une analyse contextuelle, ce qui devrait améliorer par la suite la qualité de la synthèse. Il manque également un module de traitement des abréviations, des sigles et des noms propres.

Avec la collaboration d'A. Bramoullé [**Bramoullé, 2000**], ce logiciel a été intégré dans la version 2 du dictionnaire vocal, ce qui permet, en les sélectionnant à l'aide de la souris, de synthétiser les exemples d'usage des mots du dictionnaire. Il a été également intégré dans le logiciel *Ordictée* de dictée automatique mis au point par M. Guyomard [**Guyomard, 1997**] pour les exercices de dictée où le texte français est dicté par l'ordinateur, dans le but d'en faire une version bretonne. Ce logiciel est capable de dicter au rythme de l'élève, un texte choisi par le maître ; il peut aussi détecter les fautes d'orthographe et proposer une correction.

Le dernier logiciel en cours de développement est le *correcteur d'orthographe* qui a été initié par D. Auclerc [**Auclerc, 2000**] lorsqu'il a transformé le dictionnaire vocal sous la forme d'une base de données; le logiciel recherche dans cette base les mots les plus proches lorsqu'un mot n'est pas trouvé dans le dictionnaire. Un analyseur morphologique est en cours d'intégration pour en faire un outil plus performant.

L'ensemble de ces logiciels est résumé sur la figure 1.

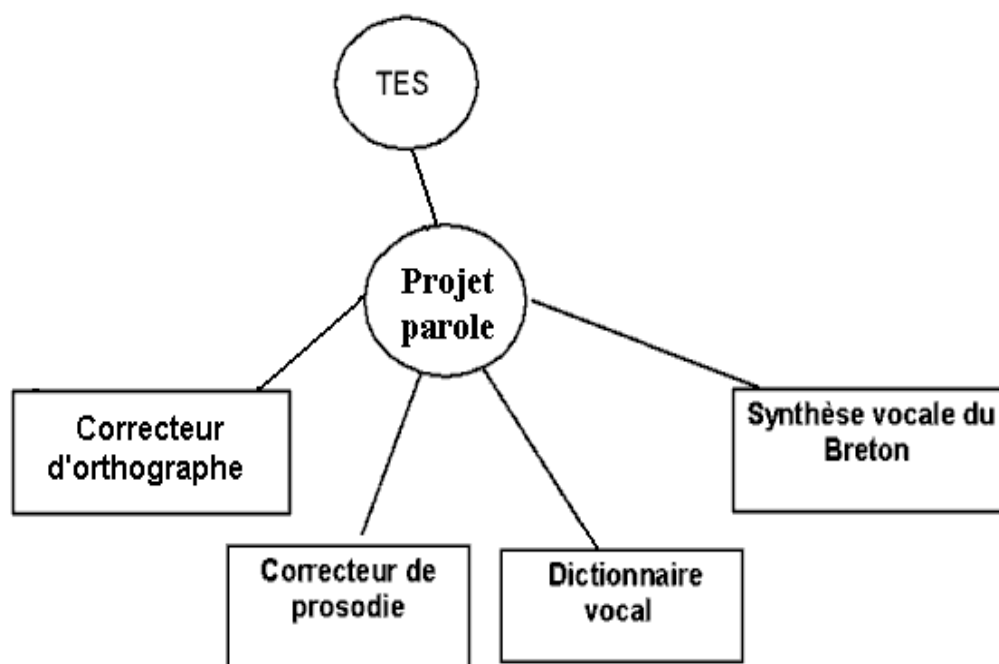


Figure 1: logiciels en cours de développement

2. Correcteur de prosodie

2.1 État des travaux

Mon travail a consisté à développer le correcteur de prosodie et à ajouter de nouvelles fonctionnalités.

A mon arrivée le correcteur (Figure 2) comprenait un module principal de pratique de prosodie, un second pour la création manuelle d'exercices de prosodie et un module de création automatique d'exercices, destiné à remplacer le module de création manuelle.

Le module de création automatique d'exercices était à peine commencé et celui de modification des frontières¹ n'existait pas.

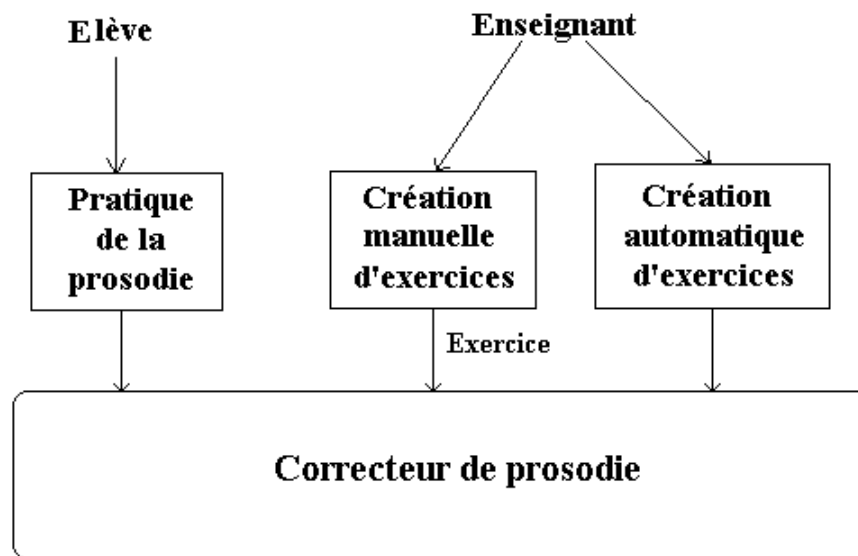


Figure 2: architecture du logiciel de correction prosodique

¹Permet de corriger manuellement les frontières des segments de paroles dans les exercices (cf. chap. Création automatique d'exercices).

2.1.1 Exercices

Un exercice est constitué d'un ou de plusieurs fichiers sonores contenant chacun une phrase d'exemple à imiter et d'un fichier d'extension *.exo* contenant le texte des phrases et leur segmentation. Ces phrases, généralement assez courtes, seront prononcées par des locuteurs maîtrisant parfaitement le breton. Quelques phrases seront fournies avec le logiciel, et d'autres pourront être enregistrées par le professeur.

Format des fichiers d'exercices

Les fichiers d'exercices créés par l'enseignant ont l'extension *.exo* et la forme suivante :

```
<Commentaire sur l'exercice (auteur, date de création, etc...)>
>
%
< Titre de l'exercice >
< Nombre de phrases >
< Description de la 1ère phrase >
%
< Description de la 2nde phrase >
%
...
%
```

Description d'une phrase:

```
< texte de la phrase >
< Nom du fichier .WAV associé >
< Description du 1er segment >
< Description du 2nd segment >
...
```

Un segment correspond le plus souvent à un phonème, mais cela peut être aussi une syllabe, un mot ou n'importe quelle portion de phrase.

Description d'un segment :

```
< Début du segment en secondes dans le signal de parole >
< Fin du segment en secondes >
< Nom du segment >
< Indice de début du segment dans la phrase >
< Indice de fin du segment dans la phrase >
```

On trouvera un exemple d'exercice dans l'annexe 1.

2.1.2 Création manuelle d'exercices

La création d'exercices consiste à segmenter et à étiqueter les phrases enregistrées par l'enseignant. Cette particularité du logiciel est l'objectif majeur: pouvoir différencier les parties importantes du signal (segments) afin de détecter les erreurs. Les phrases peuvent être segmentées en phonèmes, syllabes ou mots.

Voici l'interface du module de création manuelle d'exercices:

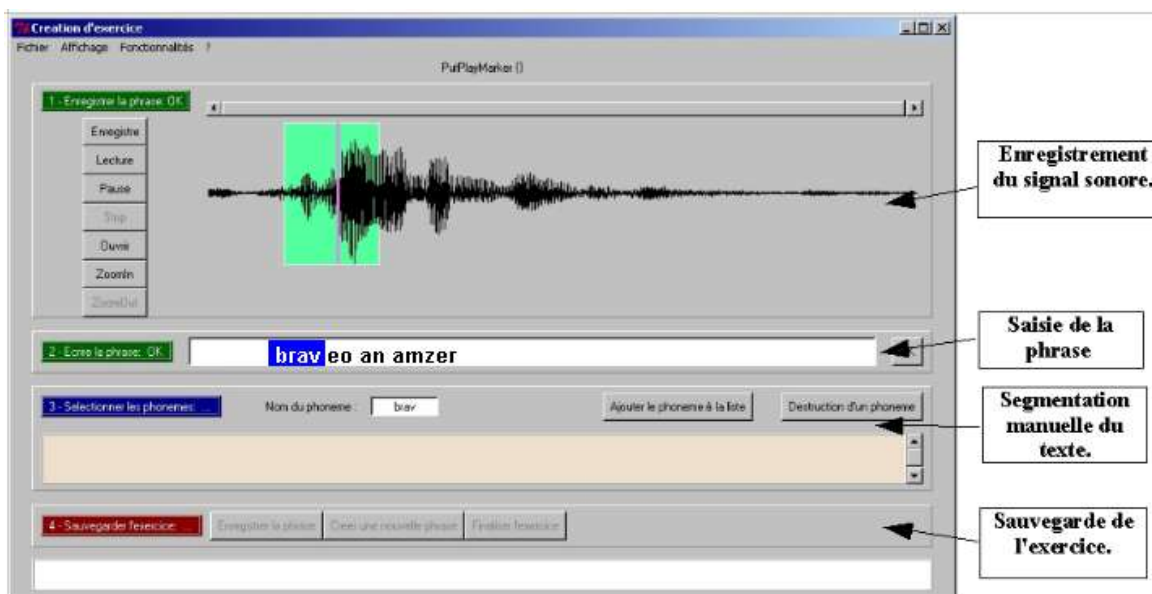


Figure 3: création manuelle d'exercices

Les étapes de la création manuelle d'un exercice sont les suivantes:

1. Choisir le fichier sonore qui servira de référence:

On peut enregistrer une phrase avec un micro en cliquant sur le bouton **Enregistre**, ou sélectionner un fichier déjà existant grâce au bouton **Ouvrir**. En général un fichier sonore contient une seule phrase d'une durée inférieure à 10 secondes.

Le bouton **Lecture** permet d'écouter le fichier sonore: si l'on est satisfait du résultat on passe à l'étape suivante, sinon on recommence.



2. Saisie du texte de l'exercice:

L'utilisateur saisit le texte de l'exercice dans la zone prévue à cet effet, puis clique sur le bouton **OK** situé à droite: ainsi dans l'exemple ci-dessous, l'utilisateur ayant enregistré la phrase "*brav eo an amzer.*" (il fait beau) écrit le texte de la phrase.



3. Segmentation manuelle:

Pour chaque segment de l'exercice, on répète les opérations suivantes:

On sélectionne d'abord une portion du signal sur la courbe sonore, puis on sélectionne le texte de ce segment dans la phrase saisie précédemment, ensuite on entre le nom du segment dans l'espace intitulé "*nom du phonème*" et pour terminer on clique sur "*Ajouter le phonème*".

Remarque:

La sélection précise d'un segment est assez difficile, surtout lorsque celui-ci n'est pas un mot: En effet, il faut sélectionner le signal, l'écouter à l'aide du bouton de lecture et recommencer jusqu'à ce que cela convienne. Afin de faciliter cette tâche on peut utiliser les fonctions de zoom, ainsi que la représentation fréquentielle du signal (le spectrogramme) qui permet avec un peu d'expérience de distinguer assez précisément les frontières des phonèmes.

4. Finalisation:

Lorsque l'étiquetage de la phrase est terminé, l'utilisateur peut ajouter d'autres phrases à l'exercice, ou terminer celui-ci en l'enregistrant et en y ajoutant éventuellement des commentaires.

Mis à part quelques problèmes mineurs que j'ai corrigés dès mon arrivée, ce module fonctionnait correctement. Cependant, la sélection manuelle étant assez fastidieuse, nous avons décidé de créer un module de "**création automatique d'exercices**" destiné à simplifier cette tâche de création.

2.1.3 Création automatique d'exercices de synthèse

La création automatique ou plutôt semi-automatique d'exercices est basée sur l'utilisation de fichiers générés par synthèse vocale à partir du texte prononcé par le maître et correspondant au signal de parole modèle. En effet, avec la synthèse vocale, on connaît exactement la durée de chaque phonème du signal de parole artificielle.

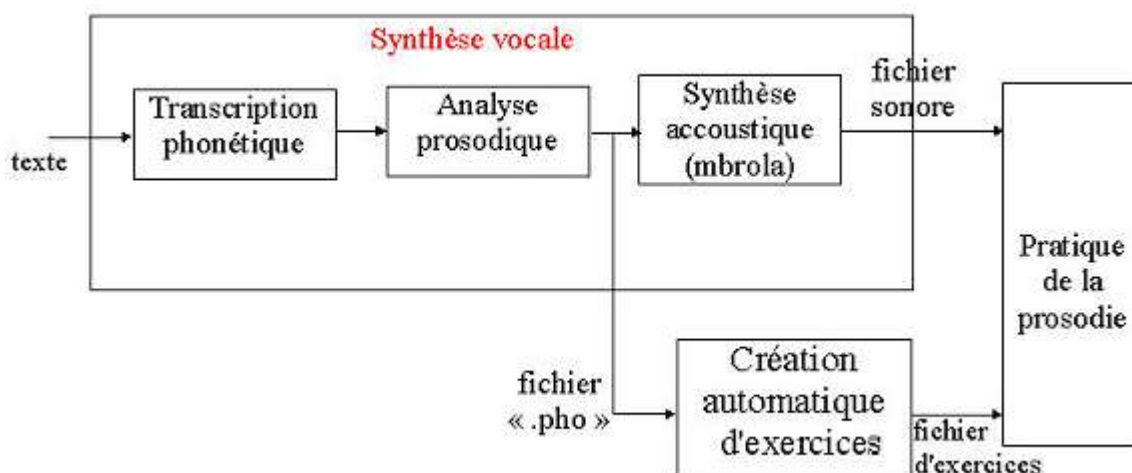


Figure 4: architecture de la création automatique d'exercices de synthèse

Comme on peut le voir sur le schéma (Figure 4), la synthèse vocale comprend trois étapes: la transcription du texte en phonétique, l'analyse prosodique et la synthèse acoustique par le logiciel *MBROLA*.

Le module d'analyse prosodique calcule la durée et la courbe de pitch (cf. glossaire, page 1) de chaque phonème. Le résultat de l'analyse prosodique est enregistré dans un fichier d'extension ".pho" que le logiciel *Mbrola* utilise ensuite pour créer le fichier sonore de synthèse.

Le fichier .pho contient toutes les caractéristiques de prosodie (durée et courbe d'intonation) pour chaque phonème du fichier sonore généré par *Mbrola*. Pour créer des exercices à partir de la synthèse, il suffit donc de récupérer les durées des phonèmes; ceci nous donne les limites de chaque segment du signal de parole artificielle généré par la synthèse, puisque le signal commence au temps $t_0 = 0$ ms.

Ayant créé un exercice de synthèse, c'est à dire un fichier "*synthese.exo*", on pourrait s'arrêter là et utiliser le signal de synthèse comme référence. Cette méthode permettrait de simplifier la création d'exercices, mais elle a un grave inconvénient: la qualité actuelle de notre synthèse vocale est très insuffisante pour servir de modèle de référence à un élève apprenant le breton. D'autre part, au début de mon projet, la synthèse vocale n'était pas intégrée au correcteur de prosodie; il fallait lancer séparément le programme de synthèse, synthétiser la phrase, l'enregistrer dans un fichier, puis dans le module de création d'exercices, taper le texte de la phrase et sélectionner manuellement les fichiers ".pho" et ".wav" correspondants.

Mon travail sur la création d'exercices a consisté à terminer la partie création automatique d'exercices de synthèse, à intégrer la synthèse au correcteur de prosodie et à ajouter une étape de comparaison maître – synthèse afin d'utiliser non plus la synthèse, mais la parole de l'enseignant comme référence dans les exercices.

2.1.4 Pratique de la prosodie

Le module de pratique de la prosodie cherche à comparer la parole de l'élève avec celle du maître. Pour faciliter cette comparaison, on peut afficher les courbes de pitch et d'énergie sur l'écran de l'ordinateur; on a donc une interface visuelle sur l'écran de l'ordinateur; l'objectif est d'obtenir une interface aussi efficace et conviviale que possible. Grâce à un algorithme de programmation dynamique, le signal de parole de l'élève est aligné automatiquement sur celui du maître.

Le schéma initial de l'interface tel qu'il se présentait au début de mon projet est représenté sur la Figure 5:

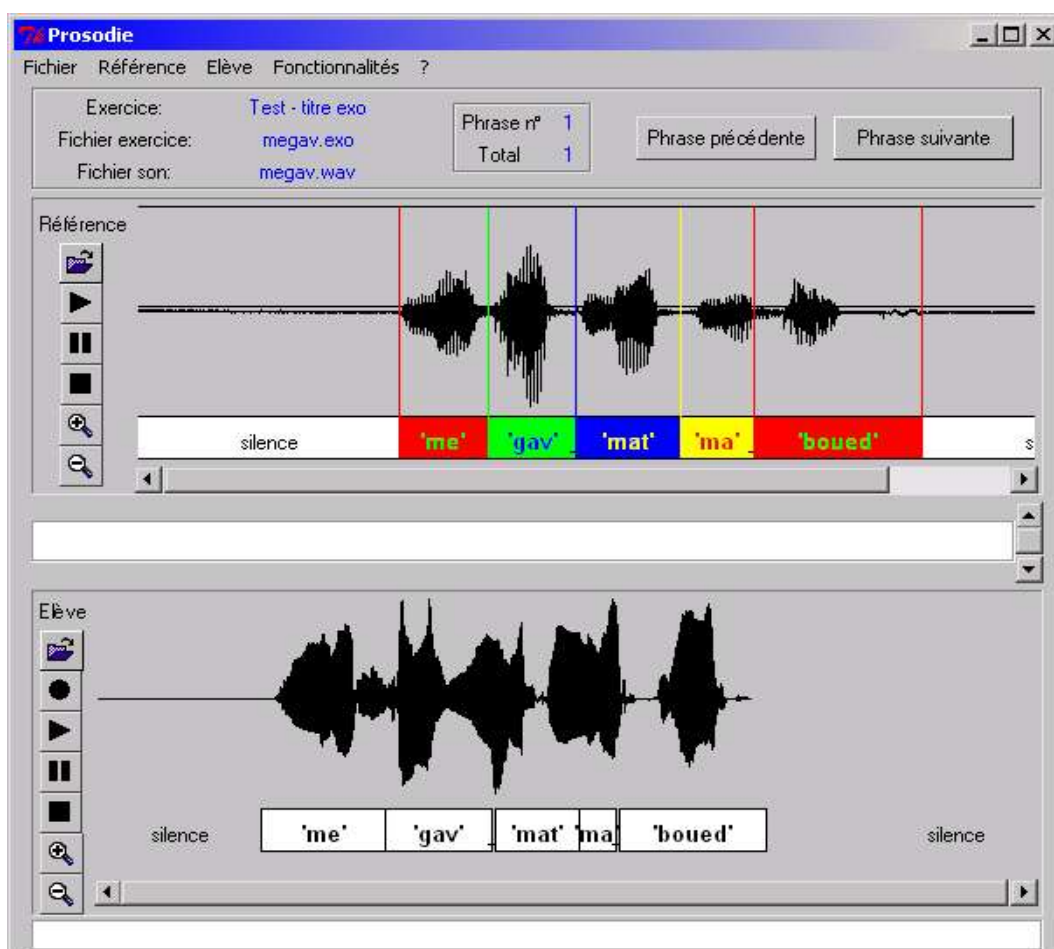



Figure 5: Interface initiale du correcteur de prosodie.


L'interface se décompose en deux parties:

- En haut la partie "maître" qui contient l'exercice, c'est-à-dire la phrase modèle étiquetée.
- En bas la partie "élève" où s'affiche la phrase de l'élève.

Au lancement du module, la partie centrale de l'interface où sont affichées les courbes et la segmentation est vide.

Pour l'utiliser, l'apprenant doit d'abord ouvrir un exercice via le menu "*Fichier*": le signal du maître apparaît dans le cadre du haut ainsi que sa segmentation.

Ensuite il s'enregistre (bouton ) en essayant de reproduire la phrase de référence. A la fin de l'enregistrement sonore, une fenêtre demande de sauvegarder celui-ci dans un fichier au format *.wav*, puis l'alignement automatique sur la phrase du maître commence.

Au lieu de s'enregistrer, on peut éventuellement sélectionner un fichier sonore (à l'aide du bouton ) enregistré précédemment.

Lorsque l'alignement automatique est terminé, la courbe du signal *élève* et sa segmentation apparaissent dans la zone d'affichage du bas.

On peut alors comparer la phrase du maître et celle de l'élève en affichant les courbes de pitch et d'énergie (à condition bien sûr que l'alignement automatique ait été correctement réalisé).

L'affichage des courbes de pitch ou d'énergie se fait respectivement dans les menus *Référence* et *élève* selon qu'il s'agit du signal de référence ou de celui de l'élève.

Mon travail sur ce module a consisté à améliorer le système d'alignement automatique et à ajouter de nouvelles fonctionnalités telles que l'affichage des courbes de pitch ou d'énergie de l'élève, dans la zone d'affichage du maître en alignant les courbes de l'élève sur celles du maître. J'ai également corrigé quelques dysfonctionnements dans l'affichage des courbes et dans la sélection simultanée de portion de signal dans les zones "*maître*" et "*élève*" et j'ai ajouté la possibilité d'afficher, pour un même exercice, la segmentation par phonèmes, par syllabes ou par mots des signaux de parole.

2.2 Choix techniques

Nous utilisons les langages C et C++ pour toutes les opérations de traitement du signal telles que l'alignement automatique, le calcul du pitch et de l'énergie. Pour la gestion de l'interface, nous aurions pu utiliser *Visual C++*, mais les outils de manipulation de fichiers sonores ou de fonctions d'analyse et de visualisation de signal sous cet environnement ne sont pas très nombreux; d'autre part ce langage de programmation n'est utilisable que sous *windows*.

Nous avons donc choisi d'utiliser le couple *Tcl/Tk* pour réaliser l'interface. *Tcl* est un langage de script¹ qui, associé à la bibliothèque graphique *Tk*, permet de réaliser très facilement des interfaces graphiques. N'étant pas compilé, le langage *Tcl/Tk* se révèle moins rapide que le C à l'exécution; c'est pour cela que nous utilisons le C pour les traitements qui réclament le plus de *ressources processeur*, et nous n'employons le *Tcl/Tk* que pour l'interface graphique.

Pour manipuler les fichiers sonores, nous utilisons *snack*². *Snack* est une librairie pour *Tcl/Tk* qui permet de lire, d'enregistrer des fichiers sonores ou encore de les afficher suivant leur représentation temporelle ou spectrale : cette librairie a été mise au point au laboratoire kth de Stockhölme.

¹ Il n'est pas compilé, il est interprété à la volée.

² <http://www.speech.kth.se/snack/>

A noter que *Tcl/Tk* et *snack* fonctionnent également sous *mac os*, *unix* ou *linux* ce qui nous permettra plus tard éventuellement de porter notre logiciel sous l'un de ces systèmes.

2.2.1 Interfaçage entre Tcl/Tk et C

Le lancement du script *Tcl/Tk* de l'interface, à partir du programme C, se fait de la manière suivante:

1. Initialisation de l'interpréteur Tcl:

```
if (Tcl_Init(interp) != Tcl_OK) {
    fprintf(stderr, "Tcl_Init failed: %s\n", interp->result);
    return Tcl_ERROR;
}
```

2. Initialisation de l'interpréteur TK:

```
if (Tk_Init(interp) != Tcl_OK) {
    fprintf(stderr, "Tk_Init failed: %s\n", interp->result);
    return Tcl_ERROR;
}
```

3. Création de lien entre une fonction C et une commande Tcl:

```
Tcl_CreateCommand(interp, "set_which", set_which, (ClientData)
0, NULL);
```

La procédure C *set_wich()* pourra être appelée depuis le script *Tcl* par la commande "*set_which*".

4. Chargement et exécution du script Tcl:

```
if (Tcl_EvalFile(interp, "choice.Tcl") != Tcl_OK) {
    fprintf(stderr, "choice.Tcl evaluation failed: %s\n",
interp->result);
    return Tcl_ERROR;
}
```

Le fichier *choice.Tcl* correspond à la fenêtre de démarrage du logiciel. Cette fenêtre contient un bouton pour chaque module du logiciel. Lorsque l'utilisateur clique sur le *n^{ème}* bouton à partir du haut, la commande "*set_which n*" est envoyé à l'interpréteur qui appelle la procédure C *set_which()* (avec comme argument *n*) qui chargera le script *Tcl* correspondant au module choisi.

A la place de la commande *Tcl_EvalFile()* on peut aussi utiliser *Tcl_Eval(interp, ChaîneTcl)* qui prend en paramètre une chaîne de caractère au lieu d'un fichier. On peut ainsi convertir les fichier *tcl* en chaîne de caractère; affecter cette chaîne de caractères à une variable *C* et l'interpréter par la commande *Tcl_Eval()*. Cela permet d'intégrer le programme C++ et tous les scripts *tcl* au sein d'un même fichier exécutable. La conversion des fichiers *tcl* en chaînes de caractères se fait à l'aide d'un script en *tcl*.

Lors du développement de l'interface, j'utilisais la fonction *Tcl_EvalFile()* afin de tester plus rapidement les modifications effectuées; mais dans la version finale tous les fichiers *tcl* seront intégrés au fichier exécutable.

Avec cette méthode il n'est pas nécessaire d'installer l'interpréteur tcl/tk sur le poste de l'utilisateur, mais seuls quelques fichiers de bibliothèques tcl/tk et snack sont copiés dans le répertoire d'installation du logiciel.

3. Comparaison et évaluation des signaux

3.1 Architecture générale

Le programme d'évaluation et de comparaison des signaux de parole se compose de plusieurs parties, réunies dans le programme C:

- Le fichier *Main.cpp*; Il fait le lien entre l'interface développée sous Tcl/Tk et la partie développée en C et C++;
- Une partie dédiée au calcul des caractéristiques prosodiques d'un signal sonore, notamment le pitch et l'énergie. Ce sont les fichiers *pit_ene_maker.cpp* et *pit_ene_maker.h*;
- Une autre partie destinée à l'alignement automatique de la parole de l'élève sur celle du maître et à la comparaison des deux signaux par l'analyse des caractéristiques cepstrales et prosodiques des signaux de parole de l'enseignant et de l'élève. Ce sont les fichiers *Diagnostic.cpp* et *Diagnostic.h*.

L'alignement automatique permet de mettre en correspondance les différents segments de parole du maître et de l'élève. Les valeurs de pitch et d'énergie mesurées sur ces segments permettent de comparer la prosodie de l'élève et celle du maître.

3.2 Calcul des caractéristiques prosodiques

Les caractéristiques prosodiques sont mesurées par l'évolution du pitch et de l'énergie et par la durée des phonèmes. La durée des phonèmes est obtenue par l'alignement automatique. Les valeurs de pitch et d'énergie sont calculées toutes les 10 ms par le programme *pit_ene_maker* composé des fichiers *pit_ene_maker.c* et *pit_ene_maker.h*. On calcule également les coefficients du cepstre qui sont utilisés pour l'alignement automatique.

3.2.1 Calcul du pitch et de l'énergie

Le programme *pit_ene_maker* calcule le pitch et l'énergie à l'aide d'un banc de filtres. Ce banc comprend 20 filtres passe-bande (de type Butterworth) et la fréquence centrale du dernier filtre est de 5 KHz. La fréquence d'échantillonnage est de 22KHz et le programme calcule les valeurs de pitch et d'énergie toutes les 10 ms. Cette partie du programme utilise la méthode décrite par J. P. Martens [Martens, 1992] de l'école polytechnique de Mons et remaniée par mes prédécesseurs pour y ajouter le calcul du cepstre.

Suppression des pics parasites:

Nous avons souvent constaté la présence de pics parasites dans les résultats obtenus par le programme de calcul du pitch. Ces pics sont de très courte durée avec des valeurs environ deux fois supérieures aux valeurs de pitch environnantes. Pour lisser les courbes de pitch et supprimer ces pics j'ai écrit la procédure "*supprimeSauts()*" qui rabaisse la valeur d'un échantillon de pitch si elle est de l'ordre du double de la mesure précédente et si elle est supérieure à la moyenne des cinq mesures environnantes de plus de 70%.

Format des fichiers de pitch et d'énergie:

Les valeurs de pitch et d'énergie sont stockées dans des fichiers du même nom que le fichier sonore correspondant et d'extension ".*pit*" pour le pitch et ".*ene*" pour l'énergie.

Le format d'un fichier ".*pit*" est le suivant :

```
Valeur maximale du Pitch (Hz) sur tout l'enregistrement
Temps1 (s)
Valeur du Pitch au temps1 (Hz)
Temps2 (s)
Valeur du Pitch au temps2 (Hz)
...
Tempsn (s)
Valeur du Pitch au tempsn (Hz)
```

Le format d'un fichier ".*ene*" est similaire à celui d'un fichier ".*pit*" :

```
Valeur maximale de l'énergie (dB) sur tout l'enregistrement
Temps1 (s)
Valeur de l'énergie au temps1 (dB)
Temps2 (s)
Valeur de l'énergie au temps2 (dB)
...
Tempsn (s)
Valeur de l'énergie au tempsn (dB)
```

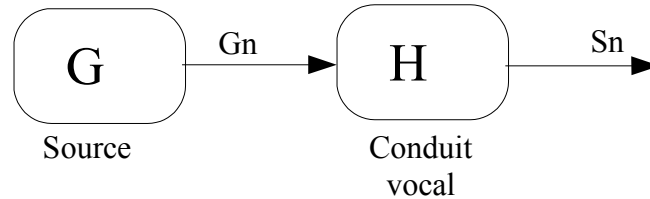
Dans les deux cas l'intervalle entre deux valeurs est toujours de 10 ms.

3.2.1 Calcul du cepstre

L'analyse cepstrale est basée sur l'hypothèse que le signal vocal S_n est produit par un signal excitateur $g(n)$ traversant un système linéaire passif (le conduit vocal) de réponse impulsionnel $h(n)$. Le signal résultant $S(n)$ est égal au produit de convolution du signal d'excitation $g(n)$ et du filtre $h(n)$:

$$s(n) = g(n) * h(n)$$

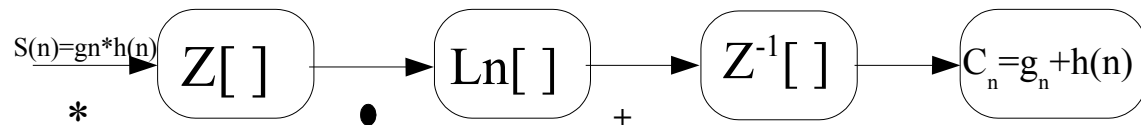
Schématiquement, le signal de parole peut être représenté par:



La source correspond au flux d'air à travers les cordes vocales; ce qui donne un signal périodique en cas de mouvement des cordes vocales et un signal bruité sinon. Par la transformée en Z , qui transforme un produit de convolution en produit, on obtient la relation:

$$S(z) = G(z).H(z)$$

Pour déconvoluer le signal et retrouver $h(n)$, on applique la relation:



Dans ce schéma, Ln est la fonction logarithmique népérien qui transforme un produit en addition, et Z^{-1} est la transformation inverse de Z .

Z est un filtre linéaire, c'est la transformée en Z , mais elle peut être remplacée par la transformée discrète de Fourier qui possède les mêmes propriétés.

Le spectre de Fourier peut être obtenu par FFT¹, par prédiction linéaire², ou par vocodeur à canaux (banc de filtres).

¹ Fast Fourier Transform: transformée de Fourier rapide

² Méthode de traitement du signal permettant de prédire la valeur du signal à l'instant n à partir des p valeurs précédentes à l'aide d'une fonction linéaire de ces p valeurs. Elle permet également de séparer dans le signal la partie due à la source de la modulation par le conduit vocal, on peut ainsi obtenir le spectre caractéristique du conduit vocal.

Pour le cepstre réel, les équations s'écrivent:

$$x(n) \xrightarrow{T.F.} X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-2j\pi nk}{N}}$$

$$X(k) \rightarrow \hat{X}_R(k) = \ln|X(k)|$$

$$\hat{X}_R(k) \rightarrow \hat{C}_n = \frac{1}{n} \sum_{k=0}^{N-1} \hat{X}_R(k) e^{\frac{-2j\pi nk}{N}}$$

T.F.: Transformée de Fourier.

n: instant d'analyse.

x(n): valeur du signal à l'instant n

N: nombre de points de la fenêtre de d'analyse.

k: fréquence à laquelle est calculée la transformée X

Lorsque le spectre est calculé à partir d'un banc de filtres distribués sur une échelle Mel¹, on peut calculer les coefficients MFCC (Mel Frequency Cepstral Coefficients) du cepstre à l'aide de la formule suivante:

$$C_n = \sqrt{\frac{2}{N}} \sum_k E_k \cos\left(\frac{\pi n}{N}(k-0.5)\right)$$

où N représente le nombre de filtres (20 dans notre cas) et E_k représente l'énergie à la sortie du filtre k. C'est la fonction *CalculateCepstre* qui calcule les coefficients du cepstre (C₁, .. C₁₀), selon cette formule, toutes les 10 ms, les énergie dans les bancs de filtres étant calculées auparavant.

Les coefficients cepstraux sont stockés dans des tableaux à deux dimensions:

```
double tabCoeffCepstreEleve[MAX_TRAMES][MAX_COEFF_CEPSTRE]; // tableau
élève
double tabCoeffCepstreEns[MAX_TRAMES][MAX_COEFF_CEPSTRE]; // tableau
maître
```

Une trame est générée toutes les 10 ms. Cela veut dire par exemple que, *tabCoeffCepstreEleve[8][j]* représente le j^{ème} coefficient de la trame au temps *t = 80ms* du signal élève.

Les coefficients du cepstre sont stockés dans un fichier du même nom que le fichier sonore correspondant, mais avec l'extension ".cep". Cette représentation cepstrale est

¹L'échelle Mel encore appelée Bark est une échelle où la répartition des filtres est approximativement linéaire dans les basses fréquences (jusqu'à 1Khz) et logarithmique dans les fréquences supérieures à 1Khz; cette répartition se rapproche de la répartition des fréquences au niveau de la membrane basilaire de l'oreille humaine.

très utilisée en reconnaissance de la parole car elle permet de caractériser le filtrage des sons par le conduit vocal.

3.3 Alignement automatique

L'alignement automatique sert à étiqueter le signal de parole de l'élève à l'aide des étiquettes des segments de la phrase de référence. Grâce à cet alignement, nous pouvons connaître l'instant de début et de fin de chacun des segments de la phrase prononcée par l'élève. Pour réaliser cette segmentation nous avons choisi une des méthodes de la programmation dynamique comparant les vecteurs mesurant les coefficients cepstraux des deux signaux de parole (Ces vecteurs sont calculés toutes les 10ms). Cette méthode utilise la matrice des distances locales entre ces vecteurs et permet de calculer récursivement les distances globales entre les zones du signal élève et du signal de référence en partant des instants de début de chacun des deux signaux pour aboutir aux instants de fin de ces signaux. La formule de récurrence nous permet de connaître à chaque instant t le chemin local optimal et donc de retrouver le chemin optimal à partir de la fin en passant par les chemins locaux optimaux. L'efficacité de cette méthode dépend d'un certain nombre de choix et de paramètres: choix des distances, choix des chemins locaux admissibles et de la zone de recherche du chemin optimal, etc.

Il existe une autre méthode plus puissante et plus fiable d'alignement de signaux: l'algorithme de Viterbi qui ressemble un peu à celui de la programmation dynamique, mais qui utilise les modèles de Markov cachés (H.M.M., pour Hidden Markov Models). Dans cette méthode, un mot ou une phrase référence, au lieu d'être représenté par une suite de vecteurs est représentée par un modèle de Markov, c'est-à-dire une suite d'états et de transitions entre états. A ces états et à ces transitions sont associées des probabilités: probabilité d'émettre un vecteur quand on est dans un état donné et probabilité de passer d'un état à un autre ou de rester dans un état. Un état peut par exemple correspondre à l'état de stabilité d'un phonème. Ensuite étant donné le signal prononcé par l'élève et représenté par la suite de vecteurs cepstraux, l'algorithme de Viterbi permet de rechercher et d'obtenir le chemin le plus vraisemblable, c'est-à-dire la suite des états ayant la probabilité la plus grande d'engendrer la suite des vecteurs du signal de l'élève. Comme la programmation dynamique, l'algorithme de Viterbi utilise une formule de récurrence pour obtenir les vraisemblances partielles, c'est-à-dire la probabilité maximale d'engendrer l'observation partielle (le signal de l'élève jusqu'à l'instant t) en atteignant l'état q_i à cet instant. Cette méthode permet d'obtenir un meilleur alignement mais elle est plus difficile à mettre en oeuvre car elle nécessite un corpus de parole important incluant divers enregistrements de nombreux locuteurs représentant les diverses variétés de voix, de parlers et de dialectes ainsi que des outils de segmentation automatique des enregistrements et des algorithmes d'apprentissage pour calculer de manière fiable les paramètres, c'est-à-dire les probabilités associées aux modèles de Markov. Comme nous n'avons pas de corpus de parole de taille suffisante ni les outils de segmentation et d'apprentissage à notre disposition nous avons choisi une méthode plus simple et éprouvée (utilisée dans de nombreux laboratoires) qui néanmoins donne des résultats satisfaisants.

L'algorithme de comparaison dynamique, basé sur le principe de programmation dynamique, permet de calculer une *distance globale* $D(I,J)$ entre deux signaux en tenant compte des variations de durée dans la prononciation des mêmes mots. La comparaison du signal maître (référence) et du signal élève revient alors à rechercher un *chemin optimal* W dans une grille de $I*J$ points. La Figure 6 représente la comparaison des signaux référence et élève dans la grille des $I*J$ points. Parmi tous les chemins W de la grille, l'algorithme de comparaison dynamique recherche celui qui minimise la distance globale entre la référence et l'élève. En pratique, il est exclu de rechercher le chemin de façon exhaustive en raison du temps de calcul. On fait appel à la programmation dynamique pour calculer récursivement la distance accumulée minimale pour chaque point (i,j) .

3.3.1 Distances locales

Les distances locales correspondent à la somme des différences entre les coefficients du cepstre d'une trame élève et les coefficients d'une trame de référence.

La formule de calcul employée est la suivante:

$$d_n(i, j) = \sum_{k=1}^N (i_k - j_k)^2 \cdot (N - k)$$

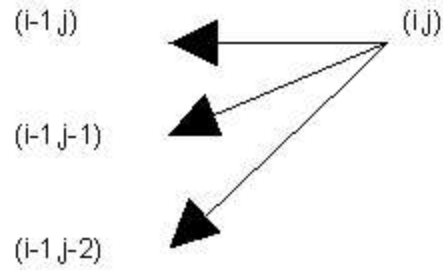
Où i_k et j_k correspondent au $k^{ième}$ coefficient cepstral de la trame de l'enseignant et de l'élève; N est le nombre de coefficients du cepstre. La pondération par $(N-k)$ permet de d'augmenter le poids des derniers coefficients du cepstre qui sont généralement plus faibles que les premiers. Ces distances sont enregistrées dans une matrice de taille égale au nombre de trames du signal de l'enseignant pour les lignes et de l'élève pour les colonnes: $d_n(i,j)$ correspond à la distance locale entre la trame i du signal de référence (maître) et la trame j du signal élève, au point $n[i,j]$ de la matrice des distances locales. Par souci de simplification $d_n(i,j)$ sera noté $d(i,j)$.

3.3.2 Distances cumulées

Les distances cumulées représentent la dissemblance entre les signaux *maître* et *élève*. Alors que les distances locales représentent la dissemblance entre les deux signaux en un instant donné, la distance cumulée en un point est la somme des distances locales depuis l'origine en suivant le chemin optimal, c'est à dire de moindre coût.

Pour préserver une certaine cohérence dans le calcul du chemin optimal, les transitions autorisées entre les points du graphe de coïncidence sont limitées à quelques uns des points les plus proches.

Ces limitations peuvent être définies par les contraintes locales suivantes que nous avons choisies :



La distance cumulée au point (i,j) est obtenue de manière récursive par la formule suivante:

$$g(i,j) = \min \left\{ \begin{array}{l} g(i-1,j) + d(i,j) \\ g(i-1,j-1) + d(i,j) \\ g(i-1,j-2) + d(i,j) \end{array} \right\}$$

La distance globale entre l'observation T et la référence R est alors donnée par:

$$D(R,T) = g(I,J) / (I + J)$$

Où I et J sont respectivement les nombres de trames des signaux R et T .

3.3.3 Chemin optimal

Chaque point de la table des distances cumulées est l'aboutissement d'un chemin de coût minimal depuis l'origine. Le chemin optimal est celui qui permet d'aboutir au point $g(I,J)$ avec le coût minimal. Pour obtenir ce chemin on part du point (I,J) ; la formule de calcul de $g(i,j)$ nous permet de trouver le point optimal précédent: $(I-1,J)$, $(I,J-1)$ ou $(I-1,J-1)$ et on procède par récurrence à partir de ce point optimal pour trouver le chemin jusqu'au point (I,I) [CALLIOPE, 1989].

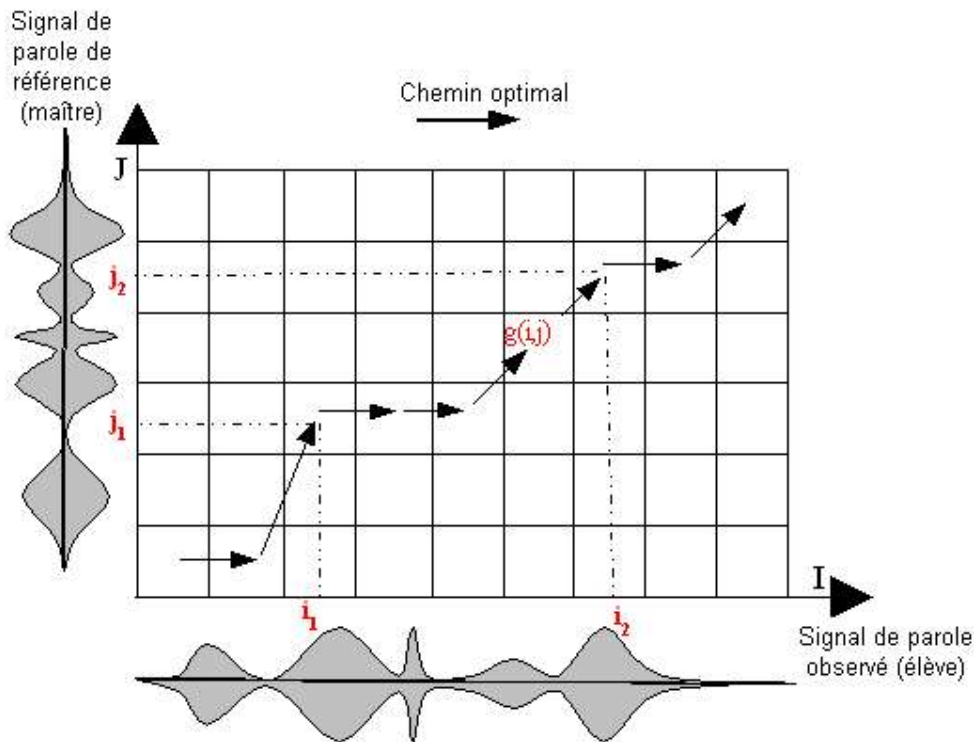


Figure 6: Table de distances cumulées et chemin optimal

Le fait que le chemin optimal passe par le point $g(i,j)$ signifie que la trame i du signal de parole observé correspond à la même portion de phrase que la trame j du signal de référence.

Concrètement le chemin optimal est sauvegardé dans le fichier "*chemin_opt.txt*"; il permet, dans le correcteur de prosodie, d'aligner la courbe du signal de l'élève sur celle de la référence et de retrouver les frontières des segments du signal élève.

Voici comment on procède pour retrouver, dans le signal élève, les frontières d'un segment compris entre les instants t_1 et t_2 pour le signal de référence:

- Sachant qu'il y a une trame toutes les 10 ms, on en déduit les numéros de trame j_1 et j_2 correspondant à t_1 et t_2 .
- On recherche dans le fichier de chemin optimal, les trames i_1 et i_2 correspondant aux trames j_1 et j_2 .
- On obtient les temps t'_1 et t'_2 délimitant le segment dans le signal élève en multipliant j_1 et j_2 par la période d'échantillonnage (10 ms).

Ces temps ainsi que les étiquettes des segments correspondants sont stockés dans un fichier *.phm* qui permettra ensuite d'afficher la phrase segmentée.

3.3.4 Détection de début et fin de parole

La présence de bruit parasite en début et en fin de parole ou pendant les pauses peut perturber l'alignement automatique; c'est pourquoi il est nécessaire de connaître les limites entre les périodes de parole et de bruit afin d'exclure les zones de bruit lors du calcul des distances et du chemin optimal. Les fonctions "*detecter_bruit_debut_eleve*" et "*detecter_bruit_fin_eleve*" ont pour rôle de déterminer le début et la fin du signal de l'élève¹. Il n'est pas nécessaire d'effectuer cette opération pour le signal de référence car les début et fin du signal sont normalement inscrits dans l'exercice.

La détection se fait grâce à la courbe de l'énergie du signal. Comme pour les mesures du cepstre et du pitch, on a calculé une mesure de l'énergie toutes les 10 ms, puis ces valeurs ont été stockées dans un fichier. Voici comment fonctionnait le système de détection de début du signal au début de mon projet:

Tout d'abord on calculait l'énergie moyenne sur l'ensemble du signal, puis on déterminait un seuil proportionnel à la moyenne obtenue:

$$\text{seuil_bruit} = \text{COEFF_SEUIL_BRUIT} * \text{energie_moyenne}$$

Le début du signal était détecté lorsque plusieurs trames (*NB_TRAMES_ENERGIE*) d'énergie consécutives étaient supérieures au seuil. Le fait d'exiger plusieurs trames successives permettait de supprimer les pics de bruits parasites (Figure 7).

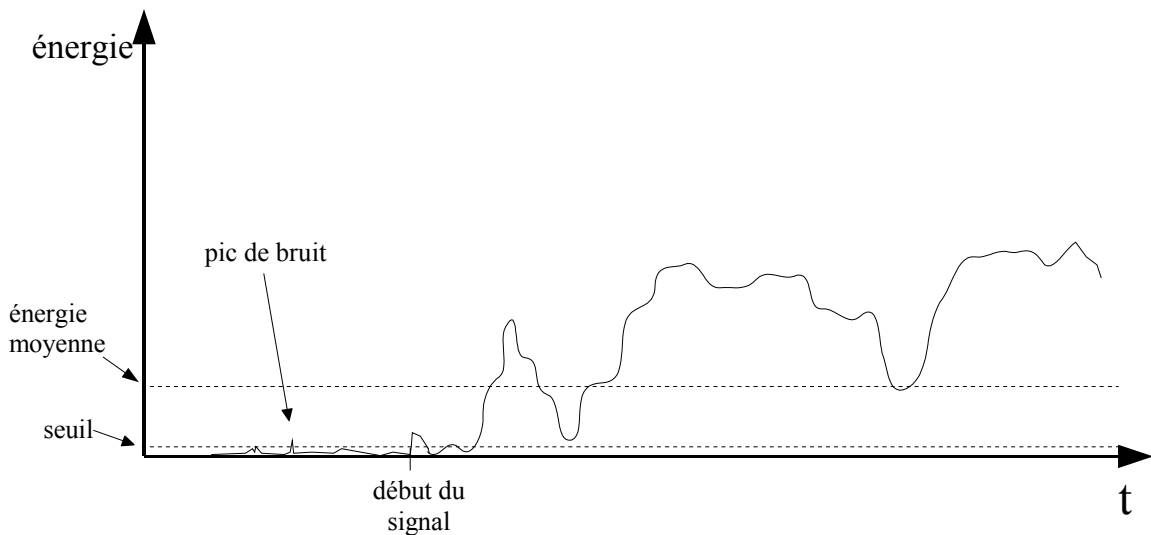


Figure 7: détection du début du signal

¹signal élève: signal à comparer avec celui qui sert de référence.

Ce premier système, assez simple, présentait quelques défauts: si la phrase était longue et les silences de début et fin très courts, l'énergie moyenne et le seuil obtenu risquaient d'être plus élevés, ce qui entraînait l'exclusion d'une partie du signal de parole. Si au contraire, la phrase contenait beaucoup de silences, le seuil était trop faible et l'on risquait d'inclure des pics de bruit dans le signal.

Pour remédier à ce problème, j'ai augmenté la valeur du seuil de bruit; ce qui permet d'obtenir une première frontière de début de signal. Ensuite on calcule l'énergie moyenne de la zone de silence obtenue, on fixe un nouveau seuil de bruit proportionnel à cette valeur moyenne et on recommence alors la recherche du début de signal, de la même manière que précédemment, mais en utilisant le nouveau seuil. L'avantage de ce second seuil est qu'il est indépendant du rapport: durée du signal / durée du silence.

3.3.5 Résultats obtenus:

L'alignement par comparaison dynamique fonctionne assez bien. Cependant il y a parfois des erreurs d'alignement. Les erreurs les plus importantes sont généralement dûes, soit à une prononciation de l'élève trop différente de celle du maître, soit à une mauvaise détection des début et fin du signal de parole.

Lorsque l'enregistrement des phrases se fait dans un environnement trop bruyant, et que le rapport signal de parole sur bruit est insuffisant, le système a du mal à différencier le signal de parole du bruit et cela fausse tous les résultats de la comparaison dynamique. Le logiciel étant destiné à être utilisé dans des classes, il faudra veiller à ce que le bruit ambiant ne soit pas trop important. Il conviendra d'utiliser de préférence des micro assez directifs qui ne seront pas trop sensibles aux bruits environnants.

On a constaté également des petites erreurs d'alignement lorsqu'on affiche la segmentation des phrases par phonème. Par exemple pour: "me" une partie du son "e" peut se retrouver dans le segment "m" obtenu par la segmentation automatique. Heureusement ces erreurs ne sont généralement visibles que lorsqu'on affiche la segmentation des phrases par phonèmes, elles sont beaucoup plus rares lorsqu'on affiche la segmentation par syllabes ou par mots.

3.4 Interface de pratique de la prosodie

Comme nous l'avons signalé précédemment, nous voulons une interface conviviale que le maître ou l'apprenant peut utiliser facilement en particulier pour comparer les différents signaux de parole et pour détecter les différences.

Voici la dernière version de l'interface de pratique de la prosodie:

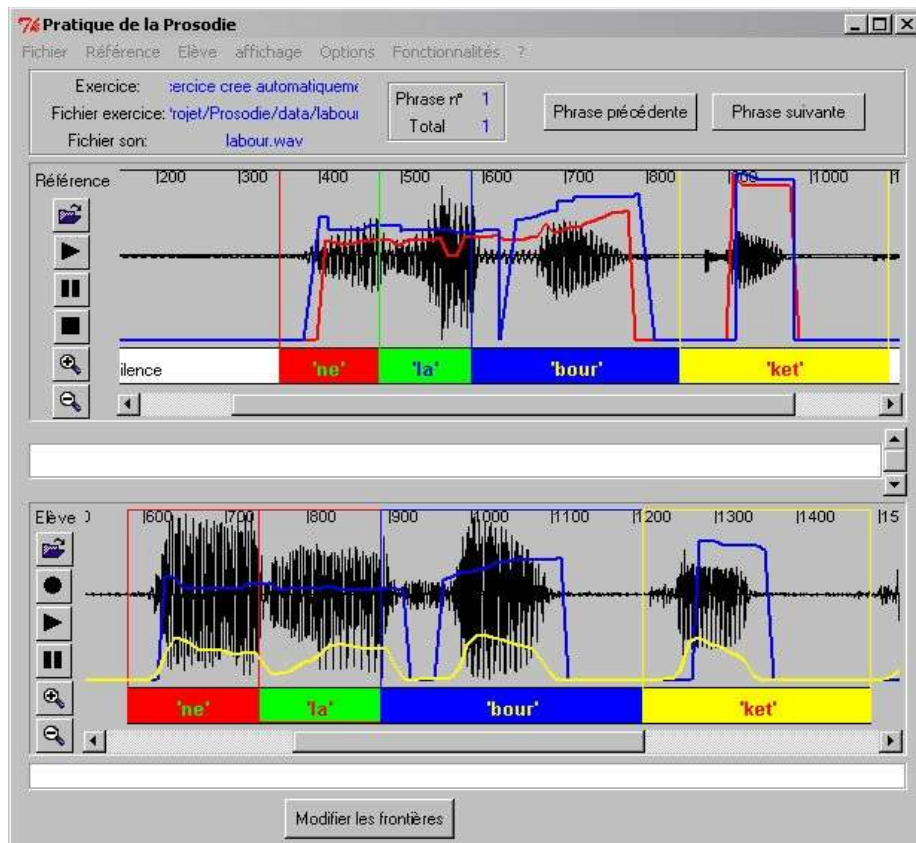


Figure 8: interface actuelle de la pratique de la prosodie.

L'utilisation de ce module a déjà été expliquée précédemment, nous ne verrons donc ici que les nouvelles fonctionnalités de l'interface.

3.4.1 Affichage des courbes sonores

Les courbes représentatives des fichiers sonores peuvent être affichées soit par "waveform", c'est à dire par l'onde représentant l'évolution du signal électrique à la sortie du microphone, ou bien sous la forme d'un spectrogramme, c'est à dire la représentation temps/fréquence du signal. Deux options *waveform* et *spectrogram* sont disponibles dans les menus "Référence" et "élève" ou dans le menu contextuel qui apparaît lors d'un clic droit dans l'une des zones d'affichage des courbes.

Voici ce qui se passe lorsqu'on clique sur le menu "waveform" du maître ou de l'élève:

- Si la courbe *waveform* est déjà affichée, elle est effacée.
- Si c'est la courbe du spectrogramme qui est affichée, elle est effacée et remplacée par la courbe *waveform*.
- Si aucune courbe n'est actuellement affichée, on affiche celle de l'onde acoustique.

Le fonctionnement est le même pour l'affichage des spectrogrammes.

Quatre variables booléennes permettent de gérer l'affichage de ces différentes courbes.

Il est également possible, dans le menu *options*, de changer la largeur de bande du spectre en modifiant la largeur de la fenêtre de *Hamming* utilisée pour le calcul du spectre. Une fenêtre étroite permet de visualiser les variations temporelles du signal, tandis qu'une fenêtre plus large permet de visualiser les différentes harmoniques. A noter que le spectre est calculé et affiché par des fonctions internes à la librairie *snack* de manière totalement transparente pour nous.

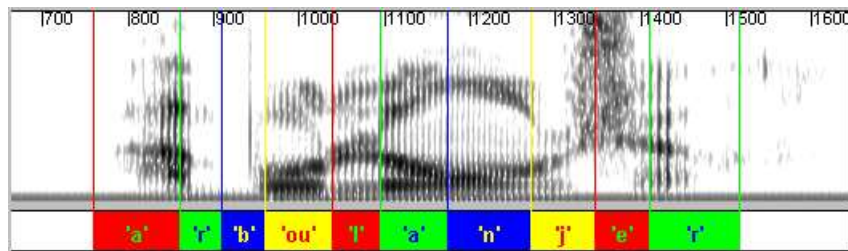


Figure 9: un spectrogramme

3.4.2 Sélection simultanée de plages du signal de parole

Lors de l'alignement automatique des signaux *maître* et *élève*, le programme crée un fichier pour représenter le chemin optimal, contenant la correspondance entre chaque trame des signaux *maître* et *élève*, c'est à dire les différents points de ce chemin. C'est ce fichier qui est utilisé pour permettre la sélection simultanée d'une fenêtre de signal dans les phrases du maître et de l'élève: lorsque l'utilisateur sélectionne une plage de signal du maître ou de l'élève, le programme recherche dans le fichier du chemin optimal, les trames correspondant aux début et fin de sélection pour l'autre locuteur.

Pour effectuer une sélection, on place le curseur de la souris dans la zone d'affichage de l'un des signaux, on presse le bouton gauche de la souris, on la déplace en gardant le bouton enfoncé et on le relâche là où l'on veut terminer la sélection. Pendant et après la sélection, des rectangles transparents de couleur (mauve si la courbe est affichée sous forme de spectrogramme, verte sinon) s'affichent sur les courbes des deux signaux (Figure 10). Si on clique sur le bouton de lecture d'un des signaux, on peut écouter le signal sélectionné; pour lire tout le signal, il faut supprimer la sélection.

Pour désélectionner les signaux, il faut appuyer sur la touche "Echap": les rectangles de couleurs disparaissent aussitôt.

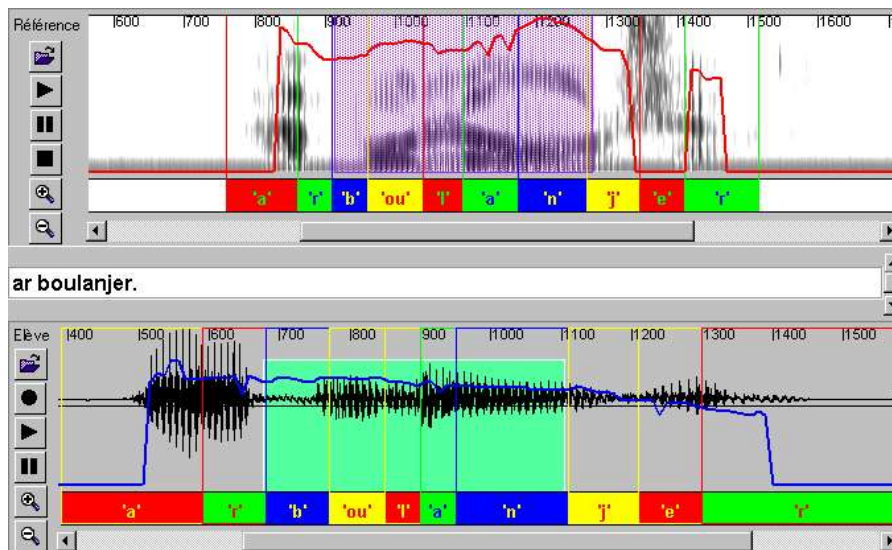


Figure 10: sélection simultanée (rectangle vert:signal de l'élève, rectangle mauve, spectrogramme de référence).

3.4.3 Affichage du pitch ou de l'énergie de l'élève dans la zone maître

Les courbes de pitch et d'énergie de l'élève peuvent être affichées dans la zone du maître et alignées temporellement sur le signal de référence. Pour cela, on utilise le fichier du chemin optimal: pour chaque point de mesure de pitch (ou d'énergie) on recherche, dans le chemin optimal, la trame correspondante dans le signal du maître et on aligne la mesure sur cette trame.

Pour comparer les courbes de pitch (ou d'énergie) du maître et de l'élève, ce qui est important ce n'est pas la valeur absolue du pitch (ou de l'énergie), mais plutôt l'évolution relative de ces courbes. Il n'est donc pas nécessaire d'employer la même échelle pour l'affichage des deux courbes qui sont superposées; pour chaque courbe l'échelle d'affichage est calculée de telle sorte que sa valeur maximale, après suppression des éventuels pics parasites, apparaisse au sommet de la zone d'affichage.

3.4.4 Affichage par phonème, syllabe ou mot

Les exercices créés par le module de création automatique permettent d'afficher différents types de segments de longueur variable: phonème, syllabe ou mot.

Les phrases des exercices sont d'abord segmentées en phonèmes et le fichier d'exercices contient la structure permettant de reconstituer la segmentation du texte par syllabes ou par mots (voir création d'exercices).

3.4.5 Indication des erreurs d'intonation

Afin de détecter les erreurs de prononciation de l'élève, on calcule pour chaque segment, une distance entre la courbe de pitch de l'élève et celle du maître. Plus la distance obtenue est élevée et plus cela signifie que l'intonation de l'élève est éloignée de celle du maître.

Pour calculer la distance de pitch on commence par diviser toutes les valeurs de pitch des deux courbes par le pitch moyen de l'ensemble de la phrase, puis on compare la valeur moyenne des segments et leur variation (montante ou descendante). Plus la valeur de distance obtenue est élevée, plus l'intonation de l'élève est mauvaise.

Lorsque la distance est inférieure à un seuil $s1$, le rectangle affiché en-dessous du segment de l'élève (figure 11) est de couleur verte, ce qui signifie que l'intonation de l'élève est correcte, au-delà d'un autre seuil $s2$ (supérieur à $s1$) le rectangle sera rouge (intonation très mauvaise) et si la distance est comprise entre les deux seuils le rectangle sera orange (intonation médiocre).

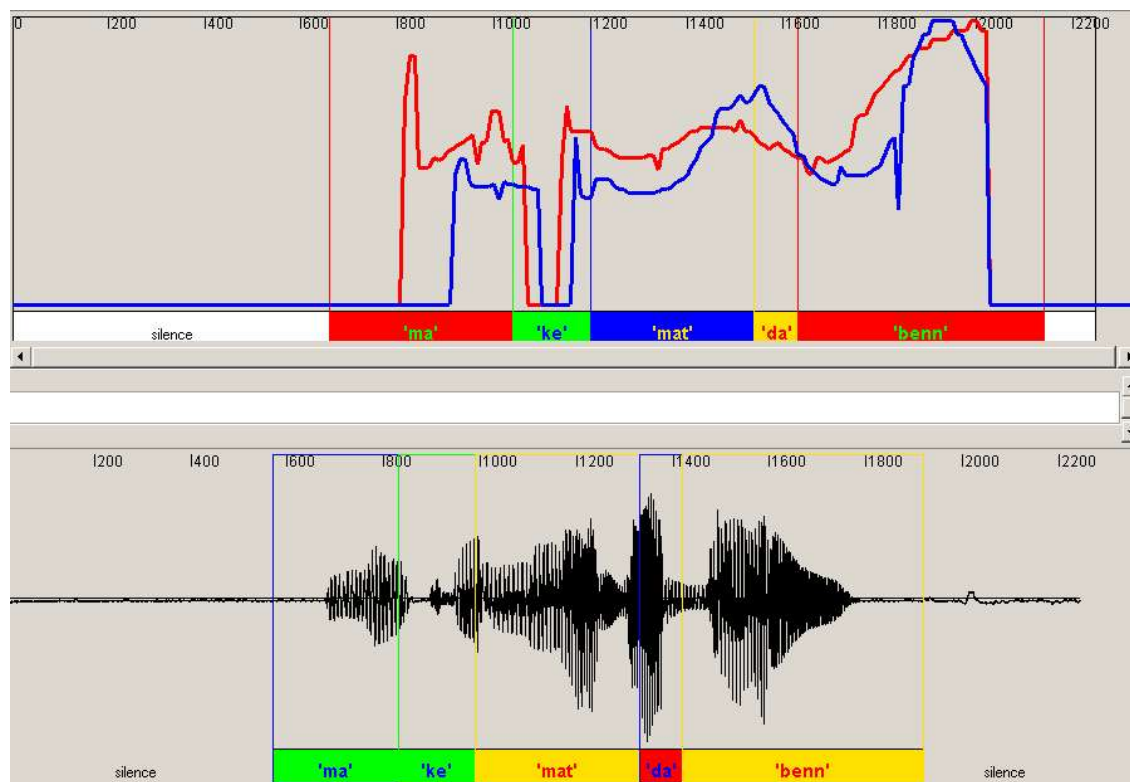



figure 11: notation de l'élève en fonction de l'intonation.

3.4.6 Autres fonctionnalités

Parmi les autres fonctionnalités ajoutées, on peut citer:

- La possibilité d'écouter les segments en cliquant sur les rectangles de couleur en bas des zones d'affichage.
- La simplification de l'enregistrement de la phrase de l'élève:
Un click sur le bouton  lance l'enregistrement, un second click sur le même bouton arrête l'enregistrement; le signal sonore est alors stocké par défaut dans un fichier *.wav* temporaire et la comparaison dynamique avec le maître est lancé automatiquement. Si l'élève veut conserver cet enregistrement sonore, il peut aller dans le menu "*Fichier -> Sauvegarder le fichier élève*", sinon ce fichier sera écrasé lors du prochain enregistrement.

4. Synthèse vocale

Le correcteur de prosodie utilise les signaux synthétisés et les fichiers phonético-prosodiques de la synthèse vocale pour étiqueter et segmenter automatiquement les signaux de parole lors de la création d'exercices; nous présentons donc dans ce chapitre le système de synthèse de parole du breton.

4.1 Principe

La synthèse vocale est la production de parole par des machines, le plus souvent à partir du texte (T.T.S.¹). Un système de synthèse vocale à partir du texte comprend généralement les étapes suivantes:

- une première phase de prétraitements chargés de remplacer les chiffres, les sigles, les abréviations, etc., par leur écriture orthographique;
- la transcription du texte en phonétique;
- l'analyse prosodique qui ajoute des marques prosodiques utilisées pour rendre la parole plus naturelle c'est à dire pour modifier l'intonation, l'accentuation et le rythme;
- un module de traitement de la parole qui transforme la séquence phonétique et prosodique en un signal de parole.

¹text to speech synthesis: synthèse vocale à partir du texte

Voici le schéma de principe de notre système de synthèse vocale pour le breton (Figure 12):

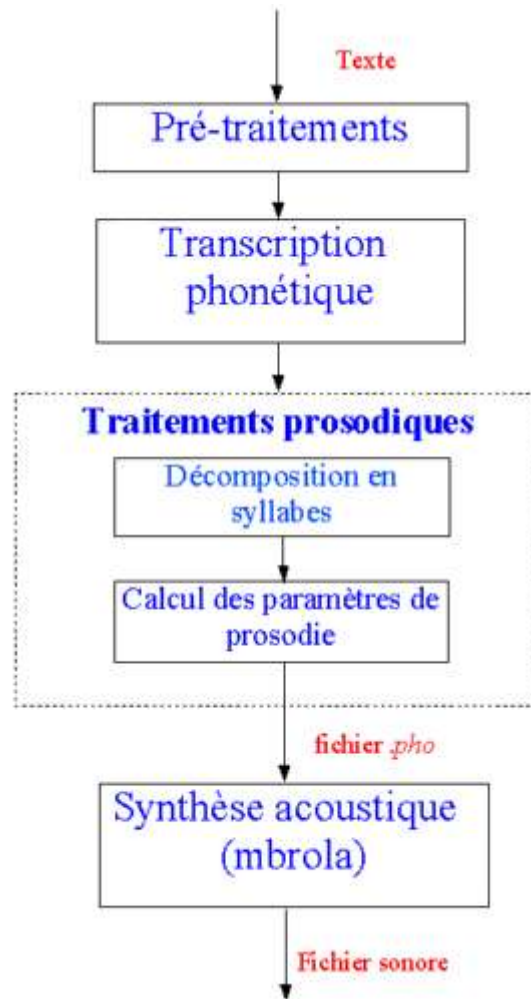


Figure 12: Architecture du système de synthèse de la langue bretonne.

4.2 Prétraitements

La première étape dans la réalisation de la synthèse d'un texte est le prétraitement qui consiste, entre autres, à remplacer les chiffres et les abréviations, par leur écriture textuelle de manière à ce qu'ils soient ensuite correctement convertis en phonèmes par le système de transcription.

Par exemple, 1543 sera remplacé par:

" pemzeg kant tri ha daou ugent "

Soit en français:

quinze cent trois et deux vingt.

Pour le moment, nous n'avons pas traité les abréviations ni les sigles; certains d'entre eux pourront être traités directement par le module de transcription graphèmes - phonèmes grâce à de nouvelles règles de transcription; les autres seront soit épelés, soit prononcés comme un seul mot.

4.2.1 Traitement des nombres

Les premiers nombres (de 0 à 20) ainsi que certains nombres simples tels que 100 ou 1000 sont directement traités par le système de transcription graphèmes - phonèmes qui se charge de les remplacer par leur écriture phonétique.

Le modules de prétraitement décompose les nombres les plus compliqués en des nombres de base qui pourront être facilement traités par le module de transcription graphèmes – phonèmes.

Voici quelques exemples de décomposition de nombres:

<i>Nombres</i>	<i>Après décomposition</i>
74	14 @ ha @ 3 20 (14 et 3 fois 20)
172	100 @ 12 @ ha @ 3 20
1345	13 100 @ 5 ha @ 40

Le symbole @ est utilisé pour indiquer au système d'ajouter de courtes pauses entre les chiffres afin de rendre la parole de synthèse plus intelligible. Les règles de décomposition des nombres sont extraites des grammaires bretonnes de F. Favereau [Favereau, 1997] et de P. Trépos [Trépos, 1994].

4.3 Transcription graphèmes - phonèmes

La transcription graphèmes - phonèmes a pour rôle de convertir le texte d'entrée en une séquence de caractères phonétiques. La transcription graphèmes - phonèmes ne donne pas toujours un phonème pour un graphème, il est même fréquent qu'un graphème donne plusieurs phonèmes.

Par exemple en français:

$x \rightarrow /k s/$ pour le mot axe, ou $/g z/$ pour le mot examen.

Un phonème comme /o/ peut provenir de plusieurs écritures (comme eau, au, aux). Enfin, la transcription phonétique d'un graphème dépend beaucoup de son contexte syntaxique dans la phrase ou encore du contexte dialectal.

Plusieurs méthodes sont couramment utilisées pour effectuer cette tâche:

- L'approche "base de données", qui consiste à stocker un maximum de connaissances phonétiques dans un lexique.
- Les systèmes à base de règles de transcription.
- Certains systèmes combinent les deux méthodes précédentes en utilisant des règles pour traiter les cas les plus courants et une base de données pour stocker les exceptions.
- Il en existe d'autres, basées sur des méthodes statistiques (réseaux de neurones, modèles de *Markov*, ou méthode par analogie).

Pour notre système de synthèse, nous utilisons actuellement un système à base de règles. Ces règles ont été créées par P. Lintanf [**An Intanv, 1994**] et reprise par J.L. Tromparent [**Tromparent, 1995**]. J'en ai rajouté quelques-unes et actuellement il y en a environ 320. Plus tard nous pourrons également combiner notre système de règles avec la base de données du dictionnaire de F. Favereau dans lequel on trouve de nombreuses variantes phonétiques et dialectales.

4.4 Génération de la prosodie

4.4.1 Principe

La transcription du texte en phonétique ne suffit pas pour obtenir une synthèse de bonne qualité. Si l'on synthétise directement le texte après la phonétisation, on obtient une parole monotone et saccadée.

L'ajout de paramètres de prosodie est donc nécessaire pour la compréhension de la parole synthétisée car cela donne à la phrase intonation, rythme et accentuation en fonction de son contenu linguistique. L'analyse prosodique a pour but de rendre plus naturelle la parole de synthèse.

La prosodie se manifeste par la variation de certains paramètres acoustiques, que l'on appelle les paramètres prosodiques décrits ci-après :

- La durée des phonèmes et des pauses.
- Le pitch, c'est à dire la fréquence fondamentale de la voix utilisée pour l'intonation de la phrase, des segments de phrase, des mots et des syllabes.
- L'énergie du signal de parole, mais nous n'utiliserons pas ce paramètre.

Dans notre système, ces valeurs sont calculées en fonction de la position des phonèmes dans les syllabes ou les mots, en fonction des catégories des phonèmes, de la ponctuation et des catégories grammaticales. Pour le moment notre modèle prosodique est très sommaire et demande à être affiné.

Prosodie des phrases :

En fonction de la ponctuation, un patron intonatif est appliqué à chaque phrase. Par exemple pour les phrases interrogatives, on a le tableau suivant :

Coefficient	1	2	3	4	5
(%)	130	110	105	110	125

La première valeur du tableau correspond au coefficient appliqué au pitch du premier mot, la seconde valeur correspond au coefficient d'un mot situé à 25% de la durée de la phrase, la suivante à 50%, et ainsi de suite. Ainsi ces valeurs nous donnent une fréquence fondamentale montant de manière sensible en fin de phrase interrogative et ayant une fréquence élevée en début de phrase. Les autres types de phrases modélisées sont les phrases affirmatives, exclamatives, les parties de phrase ponctuées par une virgule et par un point-virgule.

Décomposition en syllabes:

La décomposition des mots en syllabes permet d'appliquer ensuite des règles sur les valeurs du pitch et les durées des phonèmes. Elle s'effectue grâce à un système de règles très simples basées sur les catégories des phonèmes.

Les phonèmes sont classés en différentes catégories: voyelles, semi-voyelles, consonnes plosives, non plosives.

Accentuation :

En breton les voyelles sont généralement accentuées sur l'avant-dernière syllabe des mots. Cette accentuation se traduit par un allongement de la durée et une augmentation du pitch de la voyelle concernée. D'autre part, nous avons répertorié une liste de mots qui ne sont pas accentués (par exemple: an, ha, hag, etc.) ou le sont sur la dernière syllabe (tel que: kenavo, emañ, etc.).

4.4.2 Analyse grammaticale

On recherche les catégories grammaticales des mots dans la base de données du dictionnaire de F. Favereau. Les catégories grammaticales sont codées sous la forme d'abréviations dans les définitions des mots du dictionnaire. Pour les extraire, il a fallu écrire un programme qui extrait les abréviations et les enregistre dans des fichiers ne contenant que les noms des mots et leurs différentes catégories grammaticales. Le programme d'analyse grammaticale utilise ces fichiers pour rechercher les mots et donne en résultat la liste des catégories grammaticales correspondant aux abréviations trouvées pour chaque mot. Il est capable de retrouver la forme d'origine des mots qui ont subi une mutation*, ainsi que l'infinitif de la plupart des verbes conjugués.

* Souvent en breton, la première lettre d'un mot peut changer (muter), par exemple, le mot "ki" (chien) se transforme parfois en "c'hi" par la mutation du caractère *k* en *c'h*.

Voici le résultat de l'analyse grammaticale de la phrase: "Ar mor a laosk traezh." (La mer se retire de la plage.)

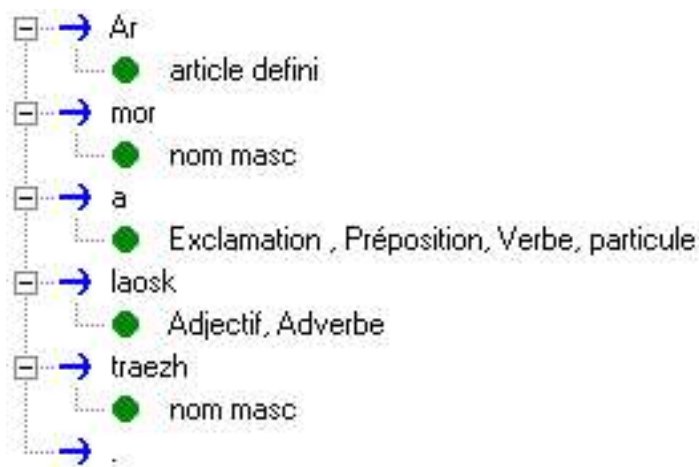


Figure 13: Résultat d'une analyse grammaticale.

Comme on peut le voir dans l'exemple de la Figure 13, le programme propose parfois plusieurs catégories grammaticales pour un même mot. Elles sont indiquées dans le dictionnaire car ce mot peut avoir plusieurs catégories grammaticales suivant le contexte où il est employé. Pour déterminer la bonne catégorie parmi celles proposées par le dictionnaire, il faudra donc rajouter à notre programme une analyse contextuelle basée sur un système de règles de grammaire.

On peut voir également dans l'exemple que le programme n'a pas su trouver que le mot "laosk" était un verbe: il a trouvé l'adverbe "laosk" (qui signifie lâchement), mais n'a pas su déterminer que ce pouvait être aussi une conjugaison du verbe "laoskaat" (relâcher): il n'est pas parvenu à retrouver le radical du verbe "Laoskaat" qui se trouvait dans le dictionnaire.

Pour le moment les résultats de l'analyse grammaticale sont peu utilisés dans le calcul de la prosodie, mais ils permettront de trouver certains repères (mots grammaticaux, mots lexicaux) et certaines frontières entre groupes prosodiques.

4.5 Synthèse acoustique

Le système de synthèse acoustique que nous utilisons est le synthétiseur *Mbrola* de la faculté de Mons. Il utilise une méthode de concaténation de diphtonges dérivée de la méthode PSOLA. Parmi les autres méthodes de synthèse existantes, on peut citer la synthèse par formant et la synthèse par unités de longueur variables.

Le choix d'utiliser la méthode *Mbrola* a été fait en 1995 (cf. §1.3) lorsqu'a commencé le développement de la première version du dictionnaire vocal (*Ar geriadur a gomz*) car elle était peu coûteuse et facile à mettre en œuvre. Depuis, une nouvelle technique de synthèse est apparue: la synthèse par unités variables qui permet d'obtenir une parole de meilleure qualité. Nous avons actuellement un projet de collaboration

avec l'université de Bonn pour utiliser cette technique, mais nous n'avons pas encore pu le réaliser par manque de moyens financiers.

La synthèse *Mbrola* étant un raffinement de la méthode *TD-Psola*, nous présentons cette technique dans le paragraphe suivant.

4.5.1 La méthode TD-Psola

La technique *TD-PSOLA* (*Time-Domain Pitch Synchronous Overlapp-Add*) permet l'application en temps réel d'un schéma prosodique sur de la parole échantillonnée. La méthode permet de modifier la fréquence fondamentale et la durée d'un segment de parole. Ces modifications sont nécessaires pour produire un signal de parole compatible avec les consignes prosodiques souhaitées (valeurs de pitch et durée différentes des valeurs de pitch et de durée des diphtonges de la base); elles sont effectuées en découpant le signal en signaux élémentaires à l'aide d'une fenêtre, à chaque période de voisement de longueur supérieure à la période fondamentale (pour la parole voisée), ou toutes les 10 ms pour la parole non voisée. Le signal original est alors fragmenté en une série de signaux de courte durée. Un nouveau signal est reconstitué en recollant les fragments avec plus ou moins de recouvrement temporel (d'où le nom de la méthode). Pour abaisser la fréquence fondamentale on va écarter les périodes de voisement, et les resserrer pour augmenter la fréquence fondamentale. Pour modifier la durée, on dupliquera ou on supprimera certaines périodes.

4.5.2 Caractéristiques principales de la méthode MBROLA

La synthèse *MBROLA* utilise l'algorithme *PSOLA* pour la concaténation des diphtonges mais auparavant des traitements spécifiques sont appliqués à la base de diphtonges originale composée de diphtonges extraits de signaux de parole naturelle. La nouvelle base de données est obtenue de la façon suivante:

La parole de départ est codée à l'aide d'un modèle d'excitation multi-bande: ceci permet de réduire considérablement la taille de la base de données. Des modifications ayant pour objectif de minimiser les problèmes de différence d'amplitude, de pitch et de phase lors de la concaténation des diphtonges sont ensuite appliquées aux unités de cette base; ainsi les diphtonges sont tous codés avec un pitch constant et l'amplitude est lissée en début et fin de diphtongue pour minimiser les différences d'amplitude lors du processus de concaténation.

4.5.3 Synthèse par unités variables

La synthèse par unités de taille variable part du principe que toute manipulation du signal (par exemple les modifications de pitch et de durée dans *TD-PSOLA*) atténue la qualité de la parole. C'est pourquoi on constitue une grande base de données de parole segmentée. Pour synthétiser une phrase, on recherche d'abord si elle se trouve dans la base; si c'est le cas on la sélectionne, sinon on recherche les mots, si un mot n'est pas dans la phrase on recherche les syllabes, puis les diphtonges. A chaque fois on sélectionne la portion de signal la plus longue possible pour minimiser les manipulations

acoustiques. Si un mot est présent plusieurs fois dans la base, on sélectionne celui qui convient le mieux en fonction de différents critères tels que le contexte phonétique, morphologique et prosodique. Des coûts sont associés à ces contextes; ces coûts mesurent les différences entre par exemple la prosodie intrinsèque des unités enregistrées et la prosodie cible souhaitée ou entre le contexte phonétique de l'enregistrement et le contexte phonétique de la phrase à synthétiser [Le Meur, 1996]. Le contexte phonétique peut correspondre par exemple à la position du phonème dans le mot.

4.6 Interface du système de synthèse

Voici l'interface de notre système de synthèse vocale:

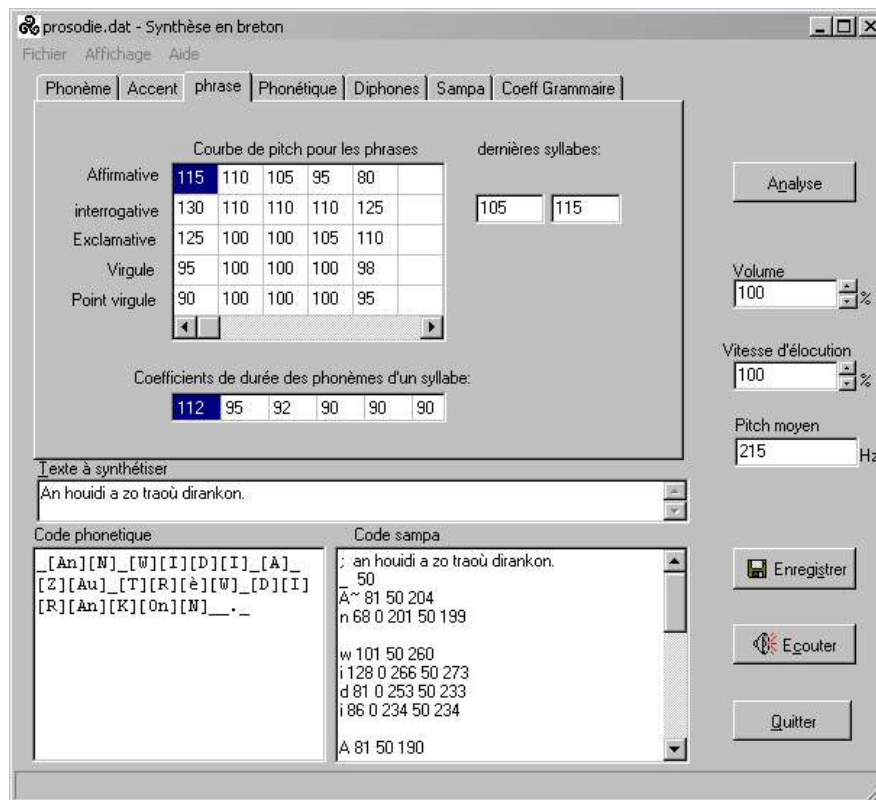


Figure 14 interface du système de synthèse.

Cette interface est uniquement destinée aux utilisateurs avertis. Elle permet de modifier les différents paramètres de calculs de la prosodie, de visualiser les résultats de l'analyse grammaticale et de la transcription en phonétique.

Utilisation:

Pour synthétiser une phrase, il faut saisir le texte dans la zone prévue à cet effet (au centre de l'interface), puis cliquer sur le bouton *Écouter* pour écouter la synthèse ou sur le bouton *Enregistrer* pour sauvegarder le résultat de la synthèse dans un fichier *.wav*.

Après la synthèse d'une phrase, le résultat de la transcription en phonétique et le code *sampa* généré pour *MBROLA* sont affichés en bas de l'interface.

La partie supérieure de l'interface contient différents onglets permettant de régler les paramètres de prosodie:

- L'onglet *phonèmes* contient la classification des phonèmes par catégories (plosives, voisées, voyelles, semi-voyelles, etc.). Cette classification permet de décomposer les mots en syllabes et d'appliquer des règles de prosodie spécifiques à certaines catégories de phonèmes.

- L'onglet *Accent* contient les coefficients à appliquer sur les durées et sur le contour de la fréquence fondamentale des phonèmes accentués ainsi que des listes d'exceptions: les mots non – accentués et les mots accentués sur la dernière syllabe.
- L'onglet *phrase* contient les coefficients de pitch à appliquer aux contours de la courbe d'intonation des phrases ou des fragments de phrase en fonction de la ponctuation.
- L'onglet *phonétique* affiche et permet de modifier le fichier de règles utilisé pour la transcription du texte en phonèmes.
- L'onglet *diphones* contient la liste des diphones absents de la base entre lesquels il faut insérer un autre phonème (le plus souvent un silence) de courte durée.
- L'onglet *Sampa* contient la table de correspondance entre les caractères phonétiques *I.P.A.** et *sampa* ainsi que leur durée par défaut.
- L'onglet *CoeffGrammaire* contient les tables de conjugaison des verbes réguliers et irréguliers utilisées pour l'analyse grammaticale.

Pour obtenir l'analyse grammaticale d'une phrase, il faut cliquer sur le bouton *Analyse* et le résultat s'affiche dans un nouvel onglet (Figure 13).

4.7 Fichiers de sortie du système de synthèse vocale

Je ne présente ici que les fichiers qui seront utilisés ensuite dans le correcteur de prosodie pour la création des exercices.

Structure des fichiers ".pho" :

Les fichiers lus par le synthétiseur Mbrola portent l'extension ".pho" et utilisent le code phonétique Sampa. Ces fichiers ".pho" reproduisent la suite des phonèmes de la phrase à synthétiser et les valeurs d'intonation et de durée à respecter; elle contient également des phonèmes "*silence*" représentant les pauses. La structure de ces fichiers est la suivante :

ex: "*me gav mad ma boued.* " (J'aime bien ce que je mange.).

```

; me gav mad ma boued.
  50
m̄ 101 50 267
e 86 0 272 50 277

  4
ḡ 101 50 258
A 86 0 264 50 270
w 83 0 261 50 252

m 101 50 248
A 171 0 252 50 257

```

*International Phonetic Alphabet: Alphabet phonétique international.

```

t 83
_ 4
m 81 50 174
A 68 0 171 50 168

b 101 50 193
w 86 0 199 50 205
E 83 0 207 50 209
t 81

_ 300

```

La structure d'un tel fichier est :

- Un commentaire est introduit par ";"
- Un silence est représenté par "_".
- Le caractère en début de ligne est un phonème en code *sampa* admis par *MBROLA*
- Le premier nombre représente la durée du phonème en millisecondes.
- Les nombres qui suivent indiquent les variations de hauteur de la voix (fréquence fondamentale).

Par exemple :

```
A 86 0 264 50 270
```

signifie que le phonème 'A' dure 86 millisecondes, et que sa fréquence fondamentale est de 264 Hertz au début et de 270 Hz au milieu du phonème (50% de sa durée).

Si l'on ouvre dans le correcteur de prosodie le fichier sonore produit par *Mbrola* pour cette phrase on obtient la courbe d'intonation suivante représentée par les variations de f_0 :

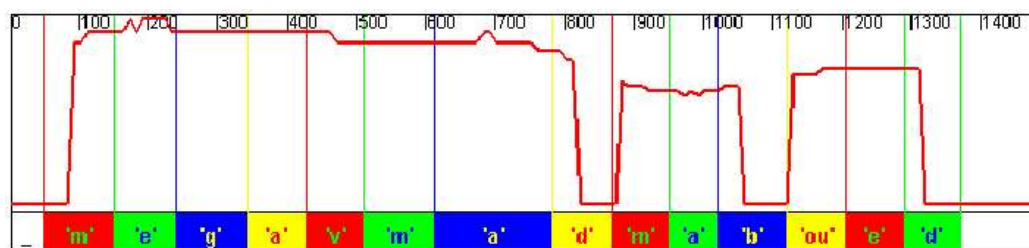


Figure 15: courbe de pitch de la phrase synthétisée: "me gav mad ma boued"

Fichiers ".phr" :

Pour la création des exercices, il est intéressant de connaître la décomposition de mots en syllabes ainsi que la correspondance entre les caractères en code *sampa* du

fichier *.pho* et les caractères du texte. Ces informations sont donc sauvegardées dans un fichier *.phr* qui aura la forme suivante pour la phrase "*Diwall rak kouezhañ !*" (*Faites attention afin de ne pas tomber !*):

```
Diwall rak kouezhañ !
//texte->phonétique->sampa:
//Syllabe
d->D->d
i->I->i
//Syllabe
w->W->w
a->A1->A
ll->L:->l
//Syllabe
r->R->r
a->A->A
k->K->k
```

Comme on peut le voir, le début de chaque syllabe est indiqué par « //Syllabe ». Les phonèmes sont quant à eux notés sous 3 formes : caractère orthographique, phonétique et *sampa*.

4.8 Imitation de la parole d'un locuteur par la synthèse vocale

A partir des fichiers générés par le système de synthèse vocale et le correcteur de prosodie, j'ai réalisé un script *tcl* qui génère un fichier (*.pho*) de synthèse imitant la parole d'un locuteur, c'est à dire avec les mêmes durées de phonèmes et la même courbe de pitch. Les fichiers nécessaires sont:

1. Un fichier *.wav* contenant la phrase à imiter.
2. La segmentation **par phonèmes** de cette phrase: soit un fichier *.exo*, soit un fichier "*.phm*" généré par la comparaison dynamique.
3. Le fichier *.pit* contenant la courbe de pitch de la phrase.
4. Un fichier de synthèse (*.pho*) de la même phrase (On peut le générer si on connaît le texte de la phrase).

Le programme récupère donc les phonèmes *sampa* dans le fichier de synthèse, lit leur durée dans le fichier *.exo* ou *.phm* (suivant que l'on imite respectivement le maître ou l'élève), obtient les valeurs de pitch dans le fichier *.pit* et génère un nouveau fichier de synthèse.

La synthèse vocale utilisant une voix de femme (celle d'Annie Ebrel), si le locuteur est un homme (voix plus grave et pitch plus bas), la parole de synthèse risque d'être beaucoup trop grave. C'est pourquoi on a fixé une valeur moyenne de pitch (220Hz), le programme calcule le pitch moyen du locuteur à imiter et en déduit le coefficient à appliquer à toutes les valeurs de pitch pour obtenir le pitch moyen voulu.

Ce programme à été intégré au correcteur de prosodie; il est accessible, via les menus "Options"->"Imiter le maître" (ou "Imiter l'élève"), dans les modules "Pratique de la Prosodie" et "Comparaison maître-synthèse".

On obtient ainsi une synthèse de bien meilleure qualité que celle produite habituellement et qui imite assez bien l'intonation du locuteur.

Remarque:

Pour obtenir de bons résultats il est indispensable que la segmentation du signal de référence soit correcte. C'est normalement le cas lorsque l'on imite le maître (on utilise le fichier d'exercice), en revanche lorsqu'on imite l'élève, la segmentation est le résultat de la comparaison dynamique avec la parole du maître et celle-ci peut faire des erreurs. Une erreur de segmentation trop importante (un phonème trop long ou trop court) décale toute la courbe de pitch, si bien que le fichier de synthèse obtenu ne ressemble plus beaucoup à la phrase du locuteur. Pour remédier à cela on peut corriger les frontières des segments manuellement (cf Modification des frontières page 58) et enregistrer la segmentation dans un nouvel exercice.

Utilité:

Cette fonction d'imitation de la parole d'un locuteur pourra plus tard être utilisée à des fins pédagogiques: on pourra par exemple imiter l'élève en exagérant les erreurs qu'il a faites, puis imiter le maître en accentuant les zones de prononciation à mettre en valeur.

Cette fonction est aussi intéressante pour l'amélioration de la qualité de notre synthèse vocale: cela permet de mettre en valeur certains défauts de la synthèse actuelle et d'en tirer de nouvelles règles de prosodie.

5. Création d'exercices

Je ne parlerai pas ici du module de création manuelle d'exercices, puisque je n'y ai pas apporté de modifications majeures, mais plutôt du module de création automatique d'exercices qui devra le remplacer.

5.1 Principe

Comme nous l'avons vu précédemment, la première version de la création automatique d'exercices ne produisait que des exercices où la synthèse vocale servait de référence. Or la qualité actuelle de la synthèse vocale du breton n'est pas suffisante pour servir de modèle à un élève. J'ai donc rajouté un module de comparaison de la parole du maître avec la parole synthétisée, qui a pour but d'étiqueter automatiquement la parole du maître, et une fonction de modification des frontières pour corriger les éventuelles erreurs de la segmentation automatique.

Voici le schéma de principe de la création d'exercices:

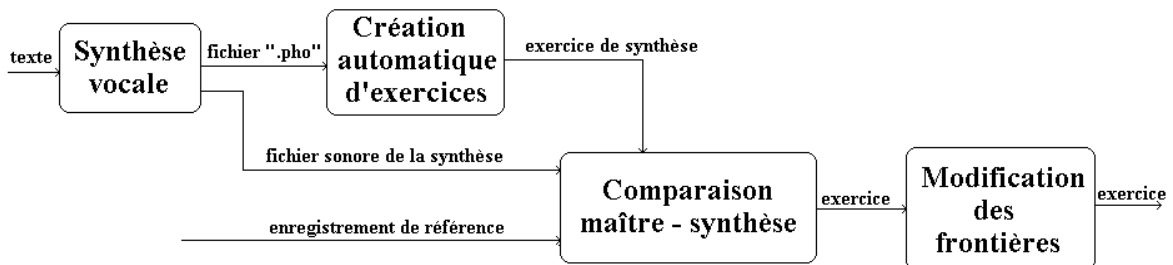


Figure 16: architecture du système de création d'exercices de prosodie.

La création d'exercices se fait suivant les étapes suivantes:

1. **Synthèse vocale** : on entre le texte de l'exercice, et on obtient un fichier *.wav* de la phrase synthétisée ainsi qu'un fichier *.pho* contenant les phonèmes de la phrase et leur durée.
2. **Création automatique d'exercice de synthèse** : le système crée alors automatiquement un exercice à partir de la synthèse, lance le module de comparaison maître-synthèse et y ouvre l'exercice de synthèse qu'il a créé.
3. **Enregistrement** : le maître s'enregistre en prononçant cette phrase.

4. **Comparaison maître – synthèse** : ce module crée un fichier d'alignement; chaque trame du signal du maître est mise en correspondance avec une trame du signal de synthèse. Le programme trouve ainsi les frontières des segments dans la phrase du maître, car on connaît les frontières des segments du signal de synthèse. On peut alors créer un exercice ayant pour modèle la phrase du maître.
5. **Modification manuelle des frontières** : comme le programme de comparaison dynamique peut faire des erreurs, l'enseignant peut corriger manuellement les frontières trouvées automatiquement.

5.2 Création automatique d'exercices de synthèse

5.2.1 Interface de création automatique

Voici comment se présente l'interface de création automatique d'exercices:



Figure 17: interface pour la création d'exercices de synthèse.

Pour créer un exercice, il faut d'abord entrer le texte de la phrase, puis on peut écouter la synthèse en cliquant sur **Ecouter** ou créer l'exercice avec le bouton **OK**.

Certains mots du breton peuvent avoir plusieurs prononciations suivant les régions, si bien que la synthèse n'emploiera pas toujours la prononciation attendue. Si la phrase prononcée par le programme de synthèse est trop différente de la phrase de référence, on risque d'avoir des erreurs lors de la phase de *comparaison maître-synthèse*.

Le menu *option* permet de lancer séparément le programme de synthèse vocale et de sélectionner manuellement le fichier *.pho* à utiliser pour la création de l'exercice. Cela permet, si l'on n'est pas satisfait du résultat de la synthèse, d'éditer le fichier *.pho*, d'y modifier les phonèmes, leur durée, et leur pitch, puis d'enregistrer le fichier et le sélectionner pour qu'il soit utilisé par le correcteur. Cette fonctionnalité est plutôt réservée aux utilisateurs avertis.

5.2.2 Intégration de la synthèse dans le correcteur

Lorsque j'ai commencé à travailler sur le projet, le programme de synthèse n'était pas encore intégré au correcteur: lorsqu'on voulait créer un exercice à partir de la synthèse, il fallait lancer séparément le programme de synthèse vocale, synthétiser la phrase, puis, dans le correcteur de prosodie, sélectionner le fichier *.pho* produit par la synthèse. Cette solution n'était évidemment pas très simple pour l'utilisateur.

Pour intégrer la synthèse vocale au correcteur, j'ai d'abord voulu utiliser une *dll*¹, mais le programme de synthèse vocale est écrit en langage *Pascal objet* sous *Delphi* et le langage *Tcl/Tk* ne supporte que les *dll* écrites en C ou C++. J'ai donc réalisé, sous *Delphi*, une version en *mode console* (c'est à dire sans interface graphique) du programme de synthèse vocale: le programme *EcouteSynth.exe*. Ce programme reprend les mêmes fichiers que le programme *synthese.exe*, mais sans les fichiers d'interface qui sont remplacés par le fichier *EcouteSynth.dpr* chargé d'interpréter les commandes et les options passées au programme.

Voici les commandes qui sont passées au programme:

- Ecouter la synthèse d'une phrase:

```
EcouteSynth.exe 1 « penaos mañ kont »
```

- Synthétiser une phrase et sauvegarder le résultat dans un fichier *.wav*:

```
EcouteSynth.exe 2 « penaos mañ kont » « synthese.wav »
```

Le premier paramètre indique au programme l'opération à effectuer. Le programme est lancé à partir d'un script *Tcl/Tk* par la commande "*exec*".

5.2.3 Création des exercices de synthèse

Une fois la phrase synthétisée, on dispose des fichiers suivants:

- Le fichier *.pho* qui contient les durées de tous les phonèmes, mais ceux-ci sont en code *sampa*.
- Le fichier *.phr* qui contient la correspondance entre les caractères *sampa* et les caractères du texte ainsi que la décomposition de la phrase en phonèmes, syllabes et mots.
- Le fichier sonore au format *.wav*.

Les informations contenues dans le fichier *.pho* nous donnent les durées des phonèmes, tandis que le fichier *.phr* nous permet de traduire les textes des segments du code *sampa* en caractères orthographiques et d'obtenir la décomposition des phrases en mots, syllabes, et phonèmes.

¹Pour "*Dynamic link library*": bibliothèque liée dynamiquement

Cela permet, à partir d'un même exercice, d'afficher la segmentation par phonème, par syllabe ou par mot.

La structure de décomposition de la phrase "ne labour ket" (il ne travaille pas) sera inscrite au début d'un exercice sous la forme suivante:

```
<STRUCT>
  <mt><sy>silence</sy></mt><mt><sy>n,e</sy></mt><mt><sy>l
,A</sy><sy>b,u,r</sy></mt><mt><sy>k,e,t</sy></mt><mt><sy>si
lence</sy></mt>
</STRUCT>
```

On utilise des balises du même style qu'en langage *xml* pour délimiter les segments:

<STRUCT> et *</STRUCT>* indiquent respectivement le début et la fin d'une phrase.

<mt> et *</mt>* " " " d'un mot.
<sy> et *</sy>* " " " d'une syllabe.

Les phonèmes d'une même syllabe sont séparés par des virgules.

La suite de l'exercice contient la liste des phonèmes, ainsi que leur temps de début et de fin.

Après quelques essais de programmation en langage *Tcl/Tk* et *Pascal* de la création d'exercices de synthèse, j'ai finalement décidé d'écrire cette partie en langage *C*. Le programme comprend la procédure *LitSampa()* qui lit le fichier *.pho*, en extrait la liste des phonèmes avec leur durée respective et place ces informations dans une liste. La procédure *LitPhr* lit dans le fichier *.phr* les informations concernant la décomposition de la phrase et la correspondance entre caractères *sampa* et textuels.

5.3 Comparaison maître-synthèse

Le module de comparaison maître-synthèse est lancé automatiquement après la création de l'exercice de synthèse.

L'interface de ce module (Figure 18) ressemble beaucoup à celle de la pratique de la prosodie, à la différence près qu'il a pour but de segmenter automatiquement la phrase du maître par comparaison avec la parole synthétisée.

La plus grande partie du code de l'interface étant identique à celui du module de pratique de la prosodie, j'ai donc réuni dans un fichier commun (nommé *commun.Tcl*) toutes les procédures communes aux deux interfaces. Les fichiers *interface.Tcl* et *maître_synthèse.Tcl* contiennent les parties distinctes des deux modules.

La Figure 18 présente le résultat de cette comparaison; dans la partie du haut on trouve la représentation spectrale du signal de synthèse avec les frontières de chaque segment. Dans le cadre du bas, on a visualisé la représentation spectrale du signal de parole du maître avec les frontières trouvées par comparaison dynamique.

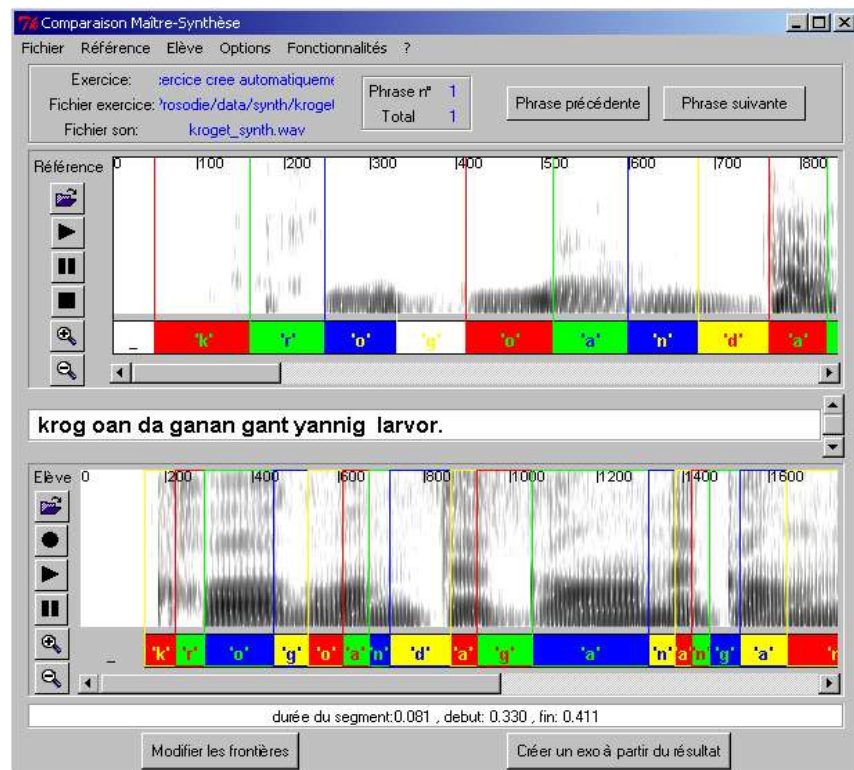


Figure 18: interface de comparaison maître-synthèse.

Si l'utilisateur est satisfait de la segmentation, il peut créer un exercice en cliquant sur le bouton "Créer un exo à partir du résultat". Sinon, la comparaison dynamique n'étant pas toujours parfaite, l'utilisateur peut, avant de créer l'exercice, modifier manuellement les frontières de segments obtenus.

5.3.1 Modification des frontières

Pour commencer la modification des frontières, il faut cliquer sur "**Modifier les frontières**". Comme son nom l'indique, ce bouton permet de corriger les frontières des segments de l'exercice. Un premier clic sur ce bouton permet de passer en mode de modification des frontières, un second clic permet de revenir au mode normal. Le bouton "**Annuler les Modifications**" permet de revenir à la segmentation d'origine.

Pour déplacer une frontière, il faut placer le pointeur de la souris sur la frontière à déplacer, presser le bouton droit de la souris, déplacer le pointeur jusqu'au nouvel emplacement de la frontière puis relâcher le bouton de la souris.

Pour obtenir une segmentation précise, il est conseillé d'agrandir le signal à l'aide du zoom. On peut également utiliser la représentation spectrale du signal qui permet de distinguer visuellement les transitions entre les phonèmes.

Lorsque l'on clique avec le bouton droit de la souris sur l'un des rectangles de couleur représentant les segments, un menu apparaît (figure 19).

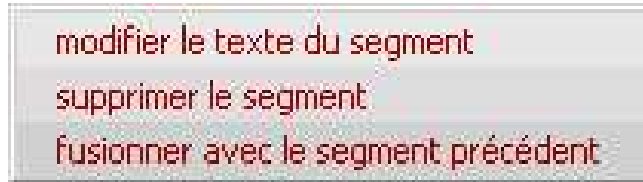


Figure 19: menu de modification des segments

Comme on peut le voir sur la figure, ce menu comporte trois entrées:

1. **Modifier le texte de segment:** qui fait apparaître une fenêtre dans laquelle on peut modifier l'étiquette du segment (celui pointé par le curseur de la souris).
2. **Supprimer le segment**
3. **Fusionner avec le segment précédent:** fusionne les segments et concatène leurs étiquettes.

Une fois que les modifications sont terminées, l'utilisateur peut créer un exercice en cliquant sur le bouton "**Enregistrer l'exercice**".

5.4 Module de modification des frontières d'un exercice

Le module intitulé "*modification des frontières*" permet, comme son nom l'indique, de modifier les frontières d'un exercice créé précédemment.

5.4.1 Utilisation

La figure 20 représente l'interface du module de *Modification des frontières*. Pour l'utiliser il faut ouvrir un fichier d'exercice à l'aide de l'icône , puis cliquer sur bouton "**Modifier les frontières**", ensuite la procédure est la même que dans le module *comparaison maître – synthèse*.

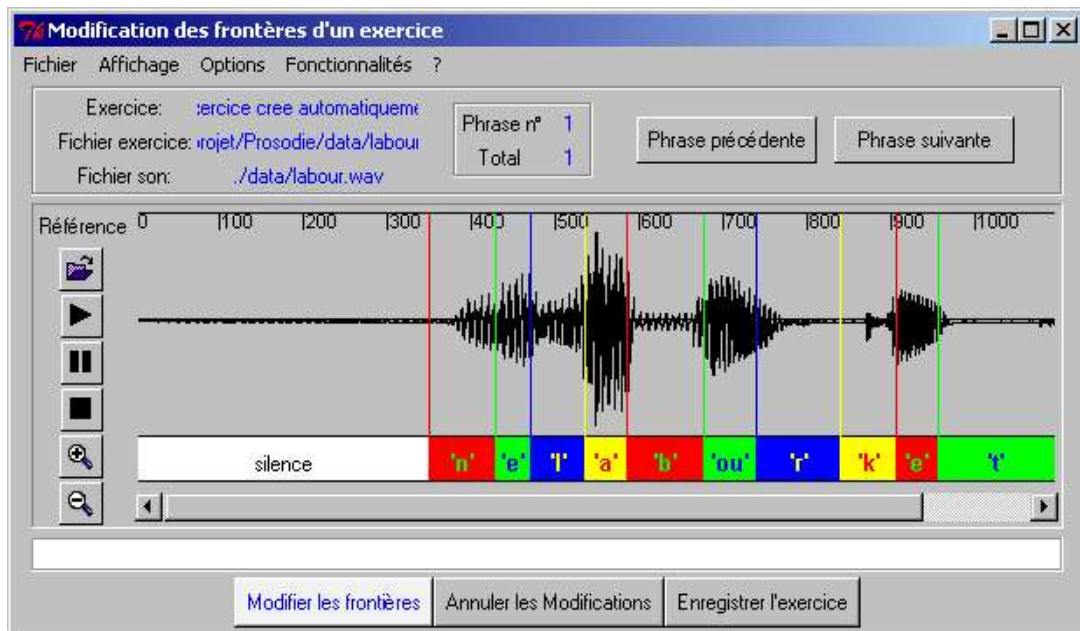


figure 20: module de modification des frontières

5.4.2 Programmation

Lorsqu'on presse le bouton "**Modifier les frontières**", les étiquettes et frontières des segments sont chargées à partir du fichier d'exercices (si l'on est dans le module de modification des frontières) ou du fichier ".phm" (si l'on est dans le module de comparaison maître-synthèse) et sauvegardées dans un tableau. Toutes les modifications sont ensuite effectuées sur les valeurs de ce tableau. Le bouton d'annulation recharge le tableau et l'affichage à partir des fichiers d'origine; le bouton d'enregistrement sauvegarde le tableau dans un fichier d'exercices.

6. Réalisation d'un premier cdrom de test

6.1 Procédure d'installation

Pour réaliser le programme d'installation j'ai utilisé le logiciel *Installshield pro*. Installer le correcteur de prosodie c'est essentiellement copier les fichiers du correcteur proprement dit et installer le système de synthèse vocale *mbrola* qui est utilisé pour la création automatique d'exercices. Le programme d'installation de *mbrola* est lancé automatiquement à la fin de l'installation du correcteur de prosodie.

6.1.1 Organisation des répertoires

Par défaut le programme d'installation placera le correcteur dans le répertoire:
"C:\Program Files\T.E.S\Correcteur de prosodie\"

Le répertoire d'installation du correcteur contient les sous - répertoires suivants:

- **Breton**, qui réunit les fichiers utilisés par la synthèse vocale lors de la création d'exercices.
- **Lib**, qui contient les librairie *Tcl/Tk* et *snack* et permet d'utiliser le correcteur de prosodie sans avoir à les installer.
- **Prosodie**, qui contient le fichier exécutable principal ("*Correcteur_de_prosodie.exe*") ainsi que le fichier d'aide du correcteur.
- **Data**, qui contient des fichiers sonores et des exercices d'exemples.



6.1.2 Les différentes étapes de l'installation

Le programme d'installation se compose des étapes suivantes:

- Choix du répertoire d'installation.
- Choix des éléments à installer:
 - Les fichiers d'exercices et l'aide en ligne sont facultatifs.
- Copie des fichiers sur le disque dur.
- Installation du logiciel *Mbrola*.

6.2 Aide en ligne

Afin de réaliser les premiers tests du logiciel auprès de quelques enseignants, j'ai réalisé une aide en ligne du correcteur. J'ai utilisé pour cela l'utilitaire *HelpWorkshop* qui est fourni avec *visual C++*.

6.2.1 Technique

Ce fichier HLP est créé à partir d'un script ".*hpj*" par le compilateur *HelpWorkshop*. Ce script .*hpj* fait référence à des fichiers Textes au format .*rtf* contenant le texte proprement dit de l'aide.

Le format RTF, pour Rich Text Format, est un format "standard" de texte développé à l'origine par Microsoft. Afin de créer un fichier RTF pour l'aide, il faut que le format RTF supporte les ajouts de notes de bas de pages. La navigation hypertexte est assurée par différentes mises en forme du fichier RTF:

1. Les notes de bas de pages
2. La navigation hypertexte
3. Codage des sauts hypertextes.

1. Les notes de bas de page

Tout d'abord chaque page de rubrique doit comporter des notes de bas de pages. Il y a quatre types de notes : les *chaînes de contexte* qui identifient une rubrique, les *titres de rubrique* qui apparaissent dans la boîte de recherche de l'aide, les *mots clés* et les *index de parcours*.

2. La navigation hypertexte

Voici résumées les quatre façons de naviguer dans les pages d'une aide :

- Les flèches de parcours << et >> quand elles sont disponibles.
- Les sauts vers une autre page.
- Les pages fugitives ou surgissantes (la page apparaît ou disparaît par un simple clic de la souris).
- Enfin on peut également naviguer avec les modules *Rechercher* et *Historique*.

sauts hypertextes de page Saut vers la page 2

pages fugitives ou surgissantes Surgissement de la page 2

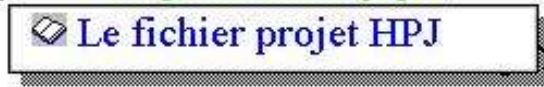


figure 21: liens hypertextes

3. Codage des sauts hypertextes.

Pour définir un saut (hypertexte) on écrit un texte doublement souligné, sur lequel on cliquera, suivi par la chaîne contexte de destination en caractères masqués:

le texte double souligné est la partie visible du saut pour l'utilisateur, le texte masqué est le code invisible de saut hypertexte...

Le texte double souligné entraîne un saut dans la même fenêtre, le texte simple souligné ouvre une fenêtre surgissante, qui disparaît par un clic souris.

Dans le fichier d'aide obtenu après compilation, les sauts de pages apparaîtront en vert souligné, les liens vers des pages surgissantes apparaîtront en vert souligné en pointillé.

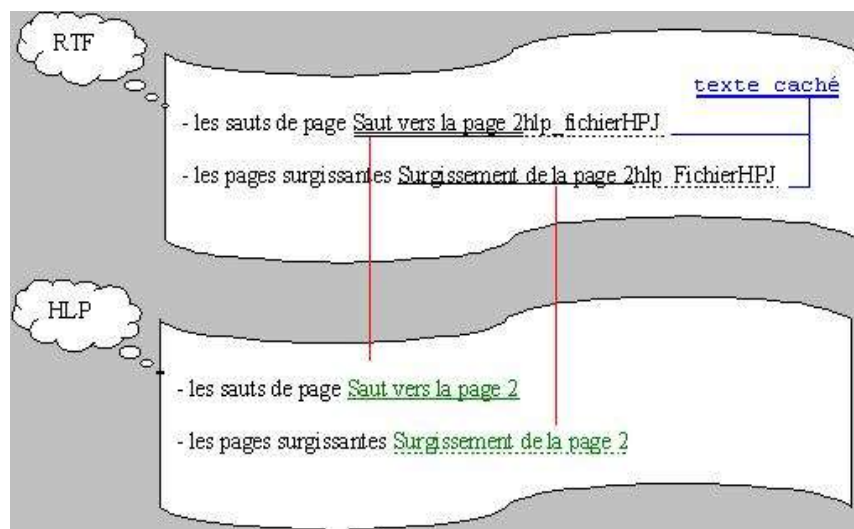


figure 22: insertion de liens hypertexte dans l'aide

6.2.2 L'aide du correcteur de prosodie

Les chapitres de l'aide sont les suivants:

- Principe
- Fenêtre de démarrage
- Pratique de la prosodie
 - Lecture du signal
 - Enregistrement de l'élève
 - Affichage des courbes de pitch et d'énergie
 - Superposition des courbes du maître et de l'élève
 - Affichage du signal
 - Sélection d'une portion du signal
 - Zoom
- La création automatique d'exercices
- Modification des frontières d'un exercice

Les deux premières pages de l'aide expliquent brièvement ce qu'est le logiciel et son utilité; les pages suivantes expliquent pas à pas, et avec de nombreuses illustrations, comment utiliser les différents modules du correcteur de prosodie.

7. Conclusion et évolution

A l'issue de ce travail de recherche, nous avons mis au point un logiciel d'apprentissage et d'enseignement de la prosodie. Dans cette étude, on a mis l'accent sur les outils de *traitement de la parole* et sur l'*interface d'utilisation* de ces outils. Les principales procédures de traitement de la parole développées et intégrées dans ce logiciel portent sur l'analyse de la parole, la synthèse de la parole en breton et la segmentation automatique.

- L'analyse spectrale (banc de filtres et calcul des coefficients du cepstre), la détection et la mesure de la fréquence fondamentale F_0 (pitch) et le calcul de l'énergie, effectués toutes les 10 ms, permettent d'obtenir une bonne représentation du signal de parole.
- La synthèse à partir du texte convertit le texte breton en phonétique à l'aide d'un ensemble de règles de transcription graphèmes-phonèmes. Un modèle prosodique simplifié permet de calculer les paramètres prosodiques : durée des phonèmes et des pauses et valeur de f_0 sur les voyelles et les consonnes voisées. La parole synthétique est produite par la technique de concaténation *Mbrola* à l'aide du répertoire de diphtongues de la langue bretonne *bzI*.
- Une procédure d'alignement automatique de la parole naturelle (prononcée par le maître ou l'élève) sur le signal de synthèse, permet de segmenter ces signaux de parole en mots, syllabes ou phonèmes, de calculer les valeurs moyennes de pitch ou d'énergie sur ces segments et d'afficher les contours de la courbe de F_0 .

Par l'intermédiaire de l'interface, le logiciel peut afficher les signaux de parole, les représentations spectrales et les courbes de pitch ou d'énergie dans deux fenêtres, l'une dédiée au maître, l'autre à l'apprenant. Des procédures permettant de sélectionner simultanément un segment de parole du maître et le segment correspondant de l'élève ont été ajoutées ainsi que des procédures d'écoute répétée de fragments de parole. Une aide en ligne complète cette interface.

Au niveau pédagogique, il n'existe pour le moment que deux possibilités pour le maître :

- soit utiliser un ensemble de mots ou de phrases segmentés en phonèmes et en syllabes pouvant servir de modèles de prononciation (en particulier au niveau de la prosodie) ou de comparaison avec les productions de l'élève ;
- soit créer lui-même ses propres mots ou ses phrases modèles ; il lui suffit d'entrer le texte écrit (le logiciel crée alors le signal de synthèse correspondant) et de parler ; le logiciel réalise la segmentation automatique de la parole du maître qui peut modifier les frontières en cas d'erreur.

La réalisation de ce travail a été facilitée par l'utilisation de *Tcl/Tk*, de *snack* et aussi des langages objet *Delphi* et *C++*.

Ce logiciel peut être utilisé tel quel dans les écoles et même par des particuliers.

Bien évidemment, ce logiciel ne correspond qu'à une première étape. Il faut maintenant, en collaboration avec les enseignants, créer de véritables leçons

pédagogiques d'apprentissage et d'enseignement de la prosodie, composées d'un ensemble d'exercices adaptés au niveau des élèves. On peut pour cela s'inspirer par exemple de ce qu'a réalisé R. Delmonte dans le logiciel *SLIM* [Delmonte, 1998] ; on pourrait créer des exercices de *perception* et de *production* portant sur l'accent des mots, avec un choix de mots typiques où l'intonation, l'intensité et la durée des syllabes accentuées sont bien marquées. On pourrait aussi créer des exercices comme cela est réalisé dans différents logiciels pour l'apprentissage de la prosodie de la phrase au niveau du rythme, des pauses et de l'évolution globale de la courbe d'intonation. Il faudra bien sûr améliorer l'interface en créant des symboles prosodiques adaptés à l'âge et au niveau de l'élève : icônes, flèches (horizontales, montantes, descendantes), schémas d'évolution prosodique plus complexes, composés de succession de symboles élémentaires.

Il faudra aussi compléter le logiciel actuel par une procédure de *détection automatique* des différences de production importantes entre la prosodie de référence et la prosodie de l'élève ; ceci pourra se faire en calculant des distances entre les courbes prosodiques sur certaines portions de signal (ou même sur toute la phrase) ou en calculant des rapports entre les paramètres de durée, d'intensité ou de pitch moyen mesurés sur des syllabes par exemple. Il faudra choisir évidemment des seuils significatifs pour déclencher une alarme et se focaliser sur les parties du signal où les différences sont significatives. En fonction du type d'erreur, il faudra prévoir des consignes adaptées permettant à l'élève d'améliorer sa prononciation ou tout au moins de se rapprocher de la prononciation souhaitée. Comme dans tout logiciel éducatif, il faudra prévoir un système d'évaluation et un historique du travail de l'élève pour quantifier ses progrès.

On peut aussi utiliser *le logiciel de synthèse de la parole* de manière indépendante comme outil sinon d'apprentissage tout au moins d'enseignement de la prosodie. En effet l'interface développée dans ce logiciel permet de modifier les paramètres prosodiques de durée et d'intonation au niveau segmental comme au niveau global. On peut ainsi montrer l'influence et l'importance de ces paramètres pour la compréhension des messages.

Il reste aussi beaucoup à faire pour améliorer et enrichir les procédures de traitement de la parole elles-mêmes. Certaines fonctions pourraient être réalisées très rapidement sans un investissement important.

Ainsi l'utilisation de la technique *TD/PSOLA* pour modifier automatiquement certaines parties du signal de l'élève (ou du maître) en calquant par exemple l'intonation ou le rythme de l'élève sur celui du maître. On peut déjà actuellement imiter la parole d'un locuteur à l'aide de la synthèse vocale (avec la voix d'Annie Ebel). Avec la technique *TD/PSOLA*, l'élève pourrait écouter sa propre parole avec la prosodie souhaitée. On peut aussi de cette manière accentuer les différences importantes : amplifier ou au contraire réduire certaines caractéristiques. Ceci peut être très utile dans des exercices de perception. Ces possibilités sont offertes dans les logiciels tels que *Winpitch* ou *Winsnorri*.

De son côté, le logiciel de synthèse doit être sérieusement amélioré et ceci à deux niveaux au moins :

1. Au niveau *segmental*, il faudra élargir la base de diphtonges actuels en utilisant des unités plus longues (syllabes et mots). Pour cela, il faudra enregistrer et segmenter de nouveaux corpus de mots et de phrases judicieusement choisis et assez importants pour pouvoir disposer de plusieurs références par unité. Le correcteur de prosodie

pourra être utilisé pour segmenter et étiqueter phonétiquement le corpus. Une coopération avec l'institut de phonétique de Bonn est en cours pour utiliser leur plateforme *Boss (Bonn Open Synthesis System)* qui permet d'une part d'extraire l'unité optimale dans un corpus et ensuite de réaliser la synthèse de la phrase.

2. Au niveau de la *modélisation de la prosodie*, il reste un travail important à réaliser aussi bien sur le rythme que sur l'intonation, en lien avec un étiquetage grammatical et une analyse syntaxique simplifiée. Les corpus précédents seront utilisés ainsi que le travail réalisé par M. Petit [**Petit, 2003**] dans son correcteur orthographique. En effet celui-ci réalise un premier étiquetage grammatical des mots de la phrase, à l'aide de la base de données du dictionnaire vocal. Il faudra aller plus loin en développant une analyse contextuelle pour désambiguïser les catégories grammaticales. Il faudra enfin concevoir un ensemble de règles permettant de modéliser les phénomènes d'accentuation et de réduction des syllabes non accentuées et les phénomènes d'intonation pour la langue bretonne. A partir de ce moment là, le logiciel de synthèse deviendra un outil bien plus intéressant pour l'enseignement et l'apprentissage de la langue bretonne (utilisation pour des exercices de dictée par exemple comme dans le logiciel ORDICTEE de M. Guyomard pour le français [**Guyomard, 1997**]).

Enfin, on ne peut ignorer que les logiciels d'enseignement et d'apprentissage de langue (*Tell me more, English Plus, Reflex'English, reflex'Deutsch, ...*) utilisent de plus en plus la reconnaissance de la parole ; avec l'aide de la reconnaissance, le logiciel peut évaluer la prononciation de l'élève, soit au niveau de la phrase soit à un niveau plus segmental [**Witt, 1998**] : mot, syllabe, phonème. La reconnaissance apporte aussi un plus pour établir des dialogues ludiques et automatiques entre l'élève et un maître artificiel. On peut encore l'utiliser pour évaluer la qualité de lecture d'un élève [**Eskenazi, 1998**]. Cependant la mise au point d'un logiciel de reconnaissance de la parole pour une nouvelle langue exige un effort important au niveau des bases de données de parole. Il faut une grande variété de locuteurs, une variété de dialectes, une grande variété de textes ; il faut ensuite segmenter et étiqueter ces corpus. L'utilisation de logiciels tels que H.T.K. [**Young, 1996**] peut faciliter cette tâche.

On ne saurait passer sous silence les possibilités qu'offrent également les visages parlants artificiels ou encore la visualisation en temps réel des articulateurs de la parole que l'on voit évoluer de façon synchrone avec la synthèse de la parole. Cette approche pourrait aider l'élève à bien positionner la langue, les lèvres, la mâchoire, etc. et ainsi l'aider à bien prononcer certains sons [**Cole, 1998; Badin, 1999**].

Mais il ne faut pas trop rêver, sachant bien que les moyens humains et financiers sont toujours très limités plus particulièrement pour les langues minoritaires ; il faudra bien sûr établir des priorités et prévoir des étapes réalistes.

8. Bibliographie

- [An Intanv, 1994] P. An Intanv, War hent fonetikadur ar Brezhoneg / Sur les chemins de la phonétisation du breton, *mémoire de maîtrise*, université de Rennes II.
- [Ar Barzh, 1996] H. Ar Barzh, corpus de parole pour la synthèse de la langue bretonne, TES/IRISA, 1996.
- [Aubry, 1999] Y. Aubry, Ordictée, logiciel de synthèse vocale en breton, *rapport de stage*, août 99, IUP MIME, Le Mans, TES/IRISA/ENSSAT, Lannion.
- [Aubry, 2000] Y. Aubry, synthèse vocale en breton, *mémoire de maîtrise* IUP MIME Le Mans, TES/ENSSAT, septembre 2000.
- [Bramoullé, 2000] A. Bramoullé, dictionnaire vocal français – breton, *rapport de projet*, TES/IRISA, ENSSAT, Lannion, mars 2000.
- [Badin, 1999] P. Badin, G. Bailly, L. J. Boë, Modèles de productions verbales et têtes parlantes virtuelles : aides utiles pour l’enseignement de la prononciation, *Speech Technology Applications in CALL*, Eurocall 99, pp. 71-76.
- [Blandin, 1999] P. Blandin, Jalons méthodologiques pour l’analyse et la conception d’activités pédagogiques en langues sur ordinateur multimédia, *mémoire de DEA en Sciences du langage* sous la direction d’Elizabeth Guimbretière, université de Rouen.
- [Calbris, 1975] G. Calbris, J. Montredon, Approche rythmique intonative et expressive du Français langue étrangère, Paris, *CLE International*.
- [Calbris, 1980] G. Calbris, J. Montredon, Oh là là, expression intonative et mimique, Paris, *CLE International*.
- [Calbris, 1986] G. Calbris, J. Montredon, Des gestes et des mots pour le dire, Paris, *CLE International*.
- [Calbris, 1992] G. Calbris, J. Montredon, Du geste à l’expression imagée, au jeu de mots et au théâtre, *Besançon Média*, université de Franche-Comté.

- [Cazade, 1999] A. Cazade, De l'usage des courbes sonores et autres supports graphiques pour aider l'apprenant en langues, <http://alsic.univ-fcomte.fr>, Vol2, N° 2, décembre 99, pp 3-32.
- [Charonnat, 1997] L. Charonnat, synthèse de la parole, *rapport de stage post-doctoral*, université de Limerick, Computer sciences & Information System department, juin 1997.
- [Charonnat, 1998] L. Charonnat, G. O Néill, G. Mercier, An Irish Speech Synthesizer, in *ESCA workshop on speech synthesis*, Jenolan Caves, Australie, nov 1998.
- [Chun, 1998] D. M. Chun, Signal Analysis software for Teaching Discourse Intonation, *Language Learning & Technology*, Vol2, N° 1, July 1998, pp. 61-77
- [Coadic 1998] R. L. Coadic, G. Mercier, J-P. Messenger, J. Siroux, La synthèse vocale de la langue bretonne, projet de correcteur de prosodie, *rapport annuel de la convention n°96-06-MDD-022-00, CEE*, septembre 1998.
- [Cole, 1998] R. Cole, T. Carmell, P. Connors, M. Macon, J. Wouters, J. de Villiers, A. Taracow, D. Massaro, M. Cohen, J. Beskow, J. Yang, U. Meier, A. Waibel, P. Stone, G. Fortier, A. Davis, C. Soland, Intelligent Animated Agents for Interactive Language Training, *ESCA – StiLL 98*, Marholmen, Suède, 24 – 27 mai, 1998, pp. 163 – 166.
- [Delcloque, 1995] P. Delcloque, The design of a French pronunciation tutor, *CALL and TELL in Theory and Practice: the proceedings of EUROCALL 1994* Rüschoff B & Wolff D; (eds),98-109.
- [Delcloque, 1999] P. Delcloque, Speech Technology Applications in CALL, *Eurocall 99*, université de Besançon.
- [Delmonte, 1998] R. Delmonte, Prosodic Modeling for Automatic Language Tutors, *ESCA – StiLL 98*, Suède, 24 – 27 mai, 1998, pp. 57 – 60.
- [Di Cristo, 1993] A. Di Christo et D. Hirst, Rythme syllabique, rythme mélodique, et représentation hiérarchique de la prosodie du français, *Travaux de l'institut phonétique d'Aix en Provence*, N° 15.
- [Dupin, 2001] J. Dupin, dictionnaire vocal multimédia français – breton, *rapport de stage*, TES/IRISA, IUP MIME Le Mans, septembre 2001.

- [**Dutoit, 1997**] T. Dutoit, An introduction to Text-To-Speech Synthesis, Kluwer Academic Publishers.
- [**Eskenazi, 1998**] M. Eskenazi, S. Hansma, The fluency Pronunciation trainer, *ESCA – StiLL 98*, Suède, 24 – 27 mai, 1998, pp. 77 – 80.
- [**Favereau, 1993**] F. Favereau, Geriadur ar Brezhoneg a-vremañ, Brezhoneg / Galleg, Galleg / Brezhoneg, Skol Vreizh, Morlaix, 1993.
- [**Favereau, 1997**] F. Favereau, grammaire du breton contemporain, Skol Vreizh, 1997.
- [**Favereau, 1999**] F. Favereau, dictionnaire usuel du breton contemporain, Skol Vreizh, 1999.
- [**Fiandino, 1999**] C. Fiandino, P. Green, A. Rouxville, Criteria for designing interactive exercises to learn French intonation, *Methods and Tool Innovations for Speech Science for Education*, Londres, 99, pp. 133-136.
- [**Finet, 2001**] S. Finet, dictionnaire multimédia français/breton, *rapport de projet ENSSAT*, université de Rennes I, mars 2001.
- [**Franco, 1998**] H. Franco, L. Neumeyer, H. Bratt, Modeling Intra-Word Pauses in Pronunciation Scoring, *ESCA – StiLL 98*, Suède, 24 – 27 mai, 1998, pp. 87 – 90.
- [**Gourmelon, 1996**] H. Gourmelon, Speech synthesis software using the TD-PSOLA method, *rapport de stage IRESTE*, université de Limerick, Computer sciences & Information System department, juin 1996.
- [**Gourmelon, 1999**] H. Gourmelon, G. Mercier, J. P. Messenger, J. Siroux, synthèse vocale en breton, *actes du colloque : le bilinguisme précoce en Bretagne, en pays celtiques et en Europe atlantique*, Klask, livre 5, PUR, Rennes, 1999, pp 125 – 138.
- [**Gros, 1984**] J. Gros, le trésor du breton parlé, le style populaire, Emgleo Breiz, Brud nevez, 1984.
- [**Guillou, 2000**] A. Guillou, Correcteur de prosodie pour la langue bretonne, *rapport de projet*, mars 2000.
- [**Guyomard, 1997**] M. Guyomard, J. Siroux, D. Pernici & C. Royer, A Speech Synthesis Application : To learn French Spelling, ELM BANK Exeter *CALL '97, Theory & Practice of Multimedia*, pp. 316-324.

- [Hamon, 1989] C. Hamon, E. Moulines & F. Charpentier, A diphone synthesis system based on time domain prosodic modifications of speech, *ICASSP, 1989*, pp.248-251.
- [Harris, 1999] N. Harris, La reconnaissance de la parole – Considérations pour son utilisation dans l'apprentissage de langues, *Speech Technology Applications in CALL, Eurocall 99*, pp. 52 – 56.
- [Hess, 2000] W. Hess, K. Stöber, Recent Development in Speech Synthesis – with special Emphasis on corpus – based Synthesis, conférence invitée, 230 WE – *Hereaus – seminar, Speech Recognition and Speech Understanding, Bad Honnef 2000*.
- [Hiller, 1993] S. Hiller et al, SPELL: An automated system for computer-aided pronunciation teaching, in *Speech Communication*, 13, pp. 463-473.
- [Humphreys, 2000] H. Humphreys, phonologie et morphosyntaxe du parler breton de Bothoa, *Ar skol Vrezhoneg, Emgleo Breizh, 1995*
- [Jilka, 1998] M. Jilka, G. Möhler, Intonational Foreign Accent : Speech Technology and Foreign Language Teaching, *ESCA – StiLL 98, Suède, 24 – 27 mai, 1998*, pp. 115 – 118
- [Keller, 2000] E. Keller, B. Zellner Keller, Speech Synthesis in Language Learning: Challenges and Opportunities, *Instill 2000*, université d'Abertay, Dundee 3 août – 2 septembre 2000, pp. 109 – 116.
- [Konopczinski, 1999] G. Konopczinski, L'acquisition du système prosodique de la langue maternelle et ses implications pour l'apprentissage d'une L2, *Speech Technology Applications in CALL, Eurocall 99*, université de Besançon, pp 62-70.
- [Langlais, 1998] P. Langlais, A. M. Oster, B. Grandström, Automatic Detection of Mispronunciation in non-native Swedish Speech, *ESCA – StiLL 98, Suède, 24 – 27 mai, 1998*, pp. 41 – 44.
- [Laprie, 1998] Y. Laprie, V. Colotte, Automatic pitch marking for speech transformations via TD-PSOLA, *Proceeding of the European Signal Processing Conference, Rhodes, Greece, 1998*, pp 1133-1136.
- [Laprie, 1999] Y. Laprie, Snorri, A software for speech sciences. *Method and Tool Innovations for Speech Science Education*, 16-17 avril 1999, University College London.

- [Larreur, 1989] D. Larreur, F. Emerard, F. Marty, linguistic and prosodic processing for text-to-speech synthesis system, *Eurospeech 1989*, pp. 510 – 513.
- [Le Meur, 1996] P. Y. Le Meur, synthèse de parole par unités de taille variable, *thèse*, université de Rennes I.
- [L’Hostis, 2002] E. L’Hostis, dictionnaire multimédia français/breton, *rapport de projet ENSSAT*, université de Rennes I, mars 2002.
- [Madigou, 1997] X. Madigou, interface graphique d’un dictionnaire vocal en breton, *rapport de projet TES/IRISA ENSSAT*, avril 1997.
- [Malfrère, 1999] F. Malfrère, T. Dutoit, Alignement Automatique du Texte sur la parole et extraction de caractéristiques prosodiques, *PhD*, Université de Mons.
- [Martens, 1992] J.P. Martens, Pitch and voiced/unvoiced determination with an auditory model.
- [Martin, 1981] Ph. Martin, Extraction de la fréquence fondamentale par intercorrélation avec une fonction peigne, *Actes des 12èmes Journées d’Etudes sur la Parole*, 1981, pp 223-232
- [Martin, 1987] Ph. Martin, Prosodic and Rhythmic Structures in French, *Linguistics*, pp. 925-949.
- [McGurk, 1976] H. McGurk et J. MacDonald, Hearing lips and seeing voices. *Nature*, 246, pp 745-746.
- [Mertens, 1990] P. Mertens, Intonation, dans *Le français parlé*, Blanche-Benveniste, Cl. Et al., éditeurs, éditions du CNRS, Paris.
- [Meador, 1998] J. Meador, F. Ehsani, K. Egan, S. Stokowski, *ESCA – StiLL 98*, Suède, 24 – 27 mai, 1998, pp.65 – 68.
- [Mercier, 1999] G. Mercier, M. Guyomard & J. Siroux, Synthèse de la parole en breton – Didacticiels pour une langue minoritaire, *Speech Technology Applications in CALL*, Eurocall 99, pp. 57 – 61.
- [Mermet, 2001] M. Mermet, an urzhiataerezh war dachenn an diwyezhegezh abred. Pleustradurioù pedagogel troet trema ar c’hehentiñ (l’ordinateur et le bilinguisme précoce, exercices pédagogiques orientés vers la communication), *mémoire de maîtrise*, juillet 2001, université de Rennes II, 95 p.

- [**Mermet, 2002**] M. Mermet, penaos deskiñ ar brosodiezh ? Implij kenaos ar gomz er skol – vamm (Comment enseigner la prosodie ? Utilisation de la synthèse de la parole dans les écoles maternelles), *rapport de DEA*, université de rennes II, juillet 2002, 128 p.
- [**Messenger, 1997**] J.-P. Messenger, Traitement de la parole et enseignement des langues, la synthèse vocale au delà du texte, *deuxième rencontres jeunes chercheurs en parole*, novembre 1997.
- [**Messenger, 1998**] J. P. Messenger, H. Gourmelon, G. Mercier, J. Siroux, Research in Speech Processing for Breton Language Training, *ESCA – StiLL 98*, Suède, 24 – 27 mai, 1998, pp. 29 – 31.
- [**Mocquard, 1999**] G. Mocquard, correcteur de prosodie, *rapport de stage* IFSIC, TES/IRISA, ENSSAT, sept. 1999.
- [**Mocquard, 2001**] C. Mocquard, Korpus prosodiezh, *mémoire de maîtrise*, université de Rennes II.
- [**Morales, 2002**] H. Morales Specian, Dictionnaire multimedia Français-Breton, *rapport de projet*.
- [**Mostow, 1999**] J. Mostow, G. Aist, Giving Help and Praise in a Reading Tutor with imperfect Listening – Because Automated Speech recognition Means never Being Able to Say You’re Certain, *CALICO Journal*, vol 16, N0 3, 1999, pp. 407 – 424.
- [**Moulet, 2000**] F. Moulet, Correcteur de prosodie, *rapport de stage licence*, IUP MIME.
- [**Moulines, 1990**] E. Moulines, F. Charpentier, Pitch synchronous waveform processing techniques for a text-to-speech synthesis using diphones, *Speech Communication*, Vol.9 (5,6), 1990, pp 453-467
- [**Pagel, 1999**] V. Pagel, De l'utilisation d'informations acoustiques supra-segmentales en reconnaissance de la parole continue, *Thèse d'université*, Université Henri Poincaré Nancy 1, 1999
- [**Parnet, 1998**] P. Parnet, Correcteur de Prosodie, *rapport de stage*, DIIC1, IFSIC, ENSSAT
- [**Petit, 2003**] M. Petit, Correcteur orthographique de langue bretonne, *rapport de projet*, ENSSAT, mars 2003 pp. 1 – 37.

- [Pruitt, 1998] J. S. Pruitt, H. Kawahara, R. Akahane-Yamada, & R. Kubo, Methods of Enhancing Speech Stimuli for Perceptual training : Exaggerated Articulation, Context Truncation, and STRAIGHT Re-Synthesis, *ESCA – StiLL 98*, Suède, 24 – 27 mai, 1998, pp. 107 – 110.
- [Rabiner, 1994] L. Rabiner, B.H Juang: Fundamentals of speech, Prentice Hall Signal Processing Series, A.V. Oppenheim editor, 1993.
- [Secrest, 1983] B. G. Secrest, G. R. Doddington, An integrated pitch tracking algorithm for speech systems , *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Boston, 1983, pp 1352-1355
- [Sevenster, 1998] B. Sevenster, G. de Krom, G. Bloothoof, Evaluation and training of second-language learners’ pronunciation using phoneme-based HMMs, *ESCA – StiLL 98*, Suède, 24 – 27 mai, 1998, pp. 91 – 94.
- [Sjolander, 1998] K. Sjolander, J. Beskow, J. Gustafin, E. Lewin, R. Carlson, B. Grandström, Web-based tools for speech technology, *Department of Speech, Music and Hearing*, KTH, Stockholm, Suède.
- [Sokol, 1996] R. Sokol, projet de synthèse vocale en breton, création du répertoire de diphones, *rapport de stage*, TES/IRISA ENSAT, juillet 1996.
- [Stöber, 2002] K. Stöber Bestimmung und Auswahl von Zeitbereichseinheiten für die konkatentative Sprachsynthese. *Sprache, Sprechen und Komputer*, band 6 (Peter Lang, Frankfurt a. M.).
- [Stöber, 2001] K. Stöber, P. Wagner, E. Klabbbers, W. Hess, Definition of a training set for unit selection based speech synthesis, in *proc. 4th ISCA Workshop on Speech synthesis* (ISCA, Bonn).
- [Sundström, 1998] A. Sundström, Automatic prosody modification as a means for foreign language pronunciation training, *ESCA – StiLL 98*, Suède, 24 – 27 mai, 1998, pp. 49 – 52.
- [Tanguy, 2000] E. Tanguy, dictionnaire vocal Gervogal breton / français, français / breton, *mémoire de licence*, IUP MIME, Le Mans TES/IRISA, lannion, septembre 2000.
- [Trebossen, 1998] Y. Trebossen, dictionnaire vocal français – breton, *mémoire de maîtrise*, TES/IRISA, IUP MIME Le Mans, juin 1998.

- [**Trepos, 1994**] P. Trepos, grammaire bretonne, *Brud Nevez*, Emgleo breizh ISBN 2-86775-134.9.
- [**Tromparent, 1995**] J. L. Tromparent, synthèse de parole en langue bretonne, transcription orthographique – phonétique, *rapport de DEA IFSIC*, université de Rennes I, 1995.
- [**Witt, 1998**] S. M. Witt, S. J. Young, Performance measures for phone-level pronunciation teaching in CALL, *ESCA – StiLL 98*, Suède, 24 – 27 mai, 1998, pp. 99 – 102.
- [**Young,1996**] S. Young, J. Jansen, J. Odell, D. Ollason, P. Woodland: The H.T.K. Book, université de Cambridge.

ANNEXES

Annexe 1

Exemple de fichier d'exercices

Voici un exercice créé par le module de création manuelle d'exercices (voir p18):

```
exo créé par le module de creation d'exo
penaos mañ kont

%
Test - titre exo
1
penaos mañ kont?
penaos.wav
0
0.485006284418
silence
-.5
-.5
0.485006284418
0.815010560414
penaos
0
6
0.815010560414
1.03626342727
mañ
7
10
1.03626342727
1.03626342727
silence
10.5
10.5
1.03626342727
1.39751810819
kont
11
15
1.39751810819
3.50004535147
silence
11.5
11.5
%
```

Voici un exercice créé par le module de création automatique d'exercices (voir p53):

Exercice créé par le module	1
de creation automatique	s
<STRUCT>	5
<mt><sy>silence</sy></mt><mt>	6
<sy>p,e</sy><sy>n,o~,s</sy></	1
mt><mt><sy>m,A~</sy></mt><mt>	1.1
<sy>k,o~,n,t</sy></mt><mt><sy	m
>silence</sy></mt>	6
</STRUCT>	7
%	1.1
Exercice créé automatiquement	1.19
1	a
penaos mañ kont.	7
./data/penaos.wav	8
0	1.19
0.39	1.3
silence	k
0	8
1	9
0.39	1.3
0.58	1.38
p	o
1	9
2	10
0.58	1.38
0.7	1.47
e	n
2	10
3	11
0.7	1.47
0.79	1.54
n	t
3	11
4	12
0.79	1.54
0.88	2.472
ao	silence
4	12
5	13
0.88	%

Annexe 2

Présentation brève du langage Tcl/Tk

Tout d'abord il faut bien distinguer les deux packages: *Tcl* qui est le langage de base supportant de l'algorithmique classique et *Tk* qui possède les données nécessaires à dessiner les interfaces

1. Tcl

Historique

Il a été conçu vers 1988 par John Ousterhout. Il est similaire (dans l'esprit) à sh (shell), csh (C-shell), Korn shell, Perl, Python...

Caractéristiques :

L'exécutable sous Windows se nomme : " **Wish** " un shell (sorte de ligne de commande) Tcl incluant Tk.

La documentation est accessible par le *shell* de Wish ou sur le Web.

C'est un langage de script de "*haut niveau*". Comme CAML il est dynamique (**interprété**). C'est un langage "universel", à usages multiples. Ce langage est extensible et enchâssable :

- **extensible** : de nouvelles fonctions, programmées en C ou en Tcl, peuvent lui être ajoutées.
- **enchâssable** : il peut être inclus à l'intérieur d'une application pour servir de langage de commande, de script.

Données en Tcl:

Représentation interne unique : Chaînes de caractères

Motivation : simplicité d'accès depuis C

représentation identique des données et des programmes

Structures de données externes :

Chaînes de caractères (« chaîne » ou [chaîne] ou {chaîne})

Nombreuses fonctions ou procédures

Listes : suite d'éléments séparés par des blancs : {une liste} de {trois éléments}

Tableaux : en fait, tables où les éléments sont désignés par des noms.

B.TK

La meilleure façon de le présenter est de montrer l'exemple classique du célèbre « Hello World ! »



Figure 6 - Un exemple simple de tk

Cette fonction affiche un bouton contenant le texte "Hello, World!". Si on clique sur le bouton, la commande correspondante est exécutée : impression d'un message sur la console, et arrêt de *wish* par destruction de la fenêtre principale.

Un *Widget* est une fenêtre. Le *gestionnaire de fenêtre* assure la présentation des fenêtres sur l'écran. Les arrangements de fenêtres sont assurés par des "*Geometry Managers*" (Ex : les commandes *pack* et *place*).

Un *Widget* peut contenir des éléments textuels, graphiques, d'autres fenêtres.

Tous les widgets supportent des options communes.

Classes:

Les *widgets* sont groupés en classes. Les classes les plus courantes sont :

- les labels → ils affichent du texte;
- les messages → ils affichent une boîte de dialogue avec un message accessible;
- les boutons → ils exécutent des commandes;
- les boutons à cocher → ils permettent de définir des options;
- les boutons radio → ils permettent de faire basculer des variables;
- les menus → ils permettent de lancer des commandes;
- les items de menus → ils permettent d'enrichir les menus (cascade, ...);

les ascenseurs verticaux ou horizontaux → ils permettent de faire défiler une fenêtre (en anglais "*scrollbar*");
les champs d'entrée → ils permettent de saisir du texte;
les *canvas* → ils permettent d'afficher des objets graphiques autres que les *widgets* (la courbe d'un signal sonore par exemple).

Autres opérations de manipulation des *widgets* :

le packer → il fait apparaître les *widgets*;
le placer → il organise les *widgets* les uns par rapport aux autres;
les liaisons → elles permettent de lier des *widgets* à des variables ou à des fonctions;
les fenêtres → elles permettent de délimiter des blocs de *widgets* parmi d'autres;
la destruction de *widgets*

C. Quelques packages Tcl utiles

- SNACK

C'est un *package* de manipulation des fichiers *audio* qu'un de mes prédécesseurs a utilisé pour l'interface du logiciel : on peut afficher, enregistrer, lire des fichiers sonores. Il a été développé à l'institut K.T.H. de Stockholm.

Annexe 3

Le code Sampa

Le code *sampa* est un pseudo code phonétique utilisé par le synthétiseur *Mbrola*. Voici la table de correspondance entre les caractères *sampa* et ceux de l'alphabet phonétique international (I.P.A.):

Phonétique	Sampa
A	A
é	e
è	E
An	A~
Anl	A~
én	E~
èn	Ě
Eu	2
æn	Ø
e	9
O	O
Au	o
I	i
Ou	u
U	y
Un	ÿ
W	w
Y	j
On	o~
In	i~
Wi	H

Phonétique	Sampa
B	b
D	d
F	f
G	g
H	h
J	Z
K	k
L	l
Lh	L
M	m
N	n
Gn	J
Ng	n
P	p
R	r
S	s
T	t
V	v
Z	z
Ch	S
X	x

Annexe 4

Règles de transcription en phonétique

Voici quelques exemples de règles utilisées dans le système de synthèse vocale lors de la transcription du texte en phonétique:

'rr' --> 'r' signifie que 'rr' doit être remplacé par 'r'

'e' --> 'é' / 'r' + 'nk' signifie que 'e' se prononce 'é' s'il est précédé d'un 'r' et suivi par 'nk'

Certaines règles utilisent des identifiants pour désigner une catégorie de caractères ou phonèmes. Les principales catégories sont:

V: voyelles

C: consonnes

P: ponctuation ou espaces

Nb: nombres

La règle suivante:

'd' --> 't'/V,C+P; signifie que la lettre 'd' se prononce 't' lorsqu'elle est précédée par une voyelle ou une consonne et suivie par une ponctuation.

Actuellement notre système compte environ 320 règles de transcription.