

Essai de création d'un générateur automatique de l'état d'annexion

MAHRAZI Mohand & NAHALI Djamel
Université de Béjaia

Résumé

Le présent travail s'inscrit dans la perspective du traitement automatique de la langue amazighe, qui consiste est un *Essai de création d'un générateur automatique de l'état d'annexion* kabyle en utilisant le logiciel C++ Builder. Il vise, d'une part, l'utilisation de l'enseignement assisté par ordinateur de l'amazighe (EAO), et d'autre part, il contribuera dans le processus de l'aménagement de la langue amazighe en disposant à cette langue un programme susceptible de générer l'état d'annexion de tous les noms amazighs (kabyles).

Mots-clés

Tamazight – Génération automatique – Traitement automatique des langues naturelles – Base de données lexicales, état d'annexion,

I- Introduction

En plus du genre et du nombre, le nom amazighe connaît la marque de l'état (état libre ~ état d'annexion) qui est une variation conditionnée par le contexte linguistique. L'opposition flexionnelle en *état d'annexion* dit aussi *état construit* vs *état libre* (ou neutre) est phénomène morphosyntaxique qui concerne une grande partie des noms (Boukhris F. et al. 2008 : 42). Selon Salem Chaker (1996 : 39), ce phénomène concerne la généralité des dialectes berbères actuels, à l'exception de certains parlers orientaux (Nefoussa, Ghadamès, Sokna, Siwa...), et du Zenaga de Mauritanie qui semblent l'avoir perdu à date récente.

Formellement l'état est lié à la voyelle initiale du nom (état libre : forme non marqué, état d'annexion : forme marquée). Le phénomène d'annexion, selon Fatima Sadiqi (1997 : 113), relève à la fois de trois domaines linguistiques : la phonologie, la morphologie et la syntaxe ; il relève de la phonologie quand il altère la prononciation de certains sons berbères ; de la morphologie quand il crée une variation flexionnelle au niveau du nom ; et de la syntaxe quand il marque le nom selon le contexte dans lequel il se trouve. Ce contexte peut être (Nait-Zerrad K., 2004 : 17) : nom en fonction de complément

référentiel (ou complément explicatif ou encore expansion référentielle selon les auteurs) correspondant à la reprise (après le prédicat) d'un nom ou d'un pronom ; nom en position de complément déterminatif ; nom après une préposition, etc.

La marque de l'état d'annexion du nom amazighe se « manifeste par une variation morphologique de la première syllabe du nom lorsque celui-ci se remplit les conditions nécessaires à la formation de l'état d'annexion » (Boukhris F. et al. 2008 : 42) et connaît un certain nombre de variantes conditionnées par l'entourage phonétique du mot et de la voyelle initiale.

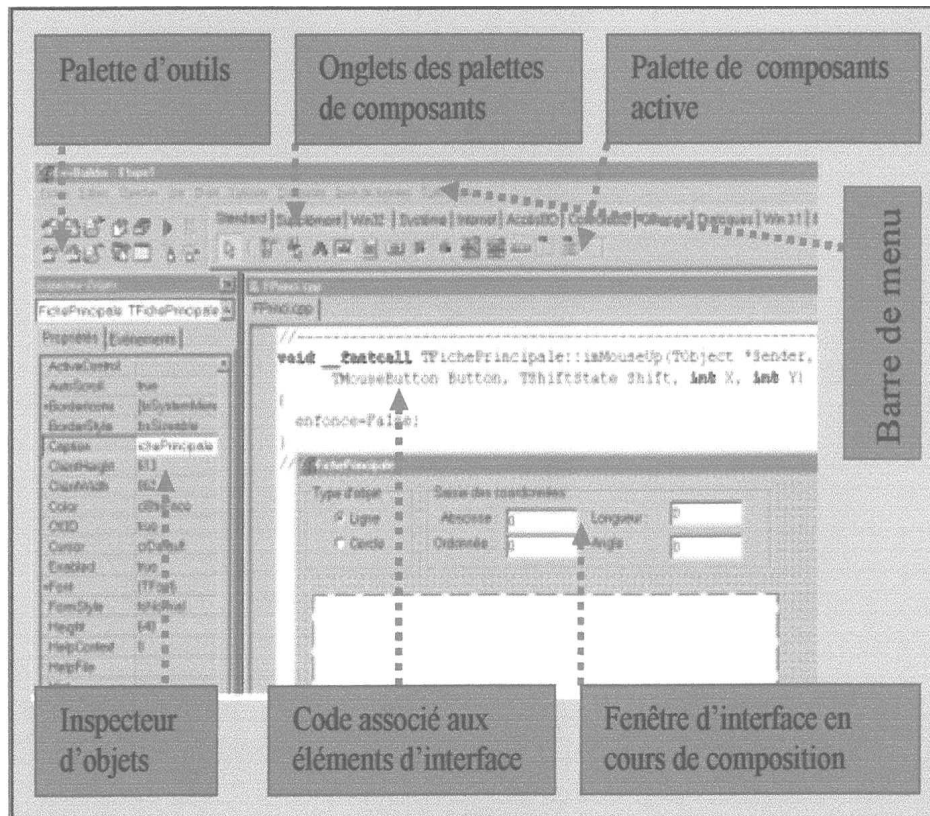
2- Le générateur

Il a pour fonction la génération de l'état d'annexion d'un nom en état libre en faisant correspondre la forme du nom à celle d'une sous catégorie de formes qui partagent les mêmes marques grammaticales et applique la règle de formation d'état correspondante. Pour ce faire, nous allons utiliser C++ Builder.

1-1- C++ Builder

C++ Builder est le nouvel environnement de développement basé sur C++ proposé par Borland. Ce dernier est un outil RAD, c'est à dire tourné vers le développement rapide d'applications (Rapid Application Development) sous Windows. Comme tout langage, il a ses règles, sa syntaxe, ses structures, ses limites, ses exceptions. Cependant, C++ Builder permet de réaliser de façon très simple l'interface des applications et de relier aisément le code utilisateur aux événements Windows, quelle que soit leur origine (souris, clavier, événement système, etc.).

2-2- L'interface de C++ Builder



Dans son interface, on peut distinguer plusieurs grandes parties :

- La barre de menu
- La barre d'outils qui se décompose en 2 grandes parties :
 - La palette d'outils permettant d'effectuer les opérations les plus courantes (sauvegarde, ouverture de fenêtres, etc.)
 - Les palettes de composants disposées accessibles par des onglets
- L'inspecteur d'objets qui permet de manipuler les propriétés des composants et d'associer du code à leurs événements.
- Les fenêtres d'interface créées par l'utilisateur. Ce sont les fenêtres de l'application en cours de création, elles portent ici le nom de *fiche*. Certaines peuvent être cachées, d'autres présentes à l'écran.

- L'éditeur de code. A chaque fiche ouverte correspond deux fichiers source (un fichier .h et un fichier .cpp) qui sont éditables dans cette fenêtre avec mise en évidence de la syntaxe.

3- Formalisation

Afin de dégager toutes ces variantes conditionnées par l'entourage phonétique du mot, nous avons procédé au dépouillement du dictionnaire de Jean-Marie Dallet *Kabyle-Français*. Les formes les plus importantes sont :

3-1- Variation de l'état d'annexion

3-1-1- Noms masculins (indépendamment du nombre)

Etat libre	Etat d'annexion	Exemple
a----	w(e) / u----	Argaz / wergaz
a----	Wa----	Asif / wasif
u----	Wu----	Udi / wudi
i----	y(e)----	Irgazen / yergazen
i----	yi----	Inifif / yinifif
i----	i----	Imawlan / imawlan

3-1-2- Noms féminins (indépendamment du nombre)

Etat libre	Etat d'annexion	Exemple
ta----	t(e)----	Tislit / teslit
ta----	Ta----	Tama / tama
tu----	Tu----	Tuga / tuga
ti----	t(e)----	Tislatin / teslatin
ti----	ti----	Tizit / tizit

3-2- Normalisation

Nous proposons un ensemble de règles qui permettent de déterminer l'état d'annexion du nom amazighe à partir de sa morphologie. Nous nous sommes inspirés des propositions faites par les enseignants du département amazigh de Béjaia¹, en essayant de déterminer l'environnement phonétique pour chaque cas. On peut les résumer ainsi :

A- Noms masculins

➤ commençant par "a" :

a → u

exceptions a → wa

- les noms génériques : *aman, allen, arrac,*
- les noms dont le pluriel garde la voyelle initiale : *azal / azalen, awal / awalen, akli / aklan ...*
- les monolitères (aC) : *agu, aḍu, ass...*
- quelques mots, comme : *asif (asif / asifen / isafen)*
→ pluriel mixte (intervation + n).

➤ commençant par "i" :

i → yi

➤ commençant par "u" :

u → wu

B- Noms féminins

➤ commençant par "ta" ou "ti" :

1e cas "ta" ou "ti": suivi d'une consonne : ta → t (*tamachut/ tmacahut, tilelli/ tlelli*),

2e cas "ta" ou "ti": suivi de 2 consonnes : ta → te (*tislit/ teslit, tamyert/ temyert*)

exceptions

- noms monolitères
ta → ta ; ti → ti ; tu → tu
- noms génériques
tamart, tazart,

➤ commençant par "tu":

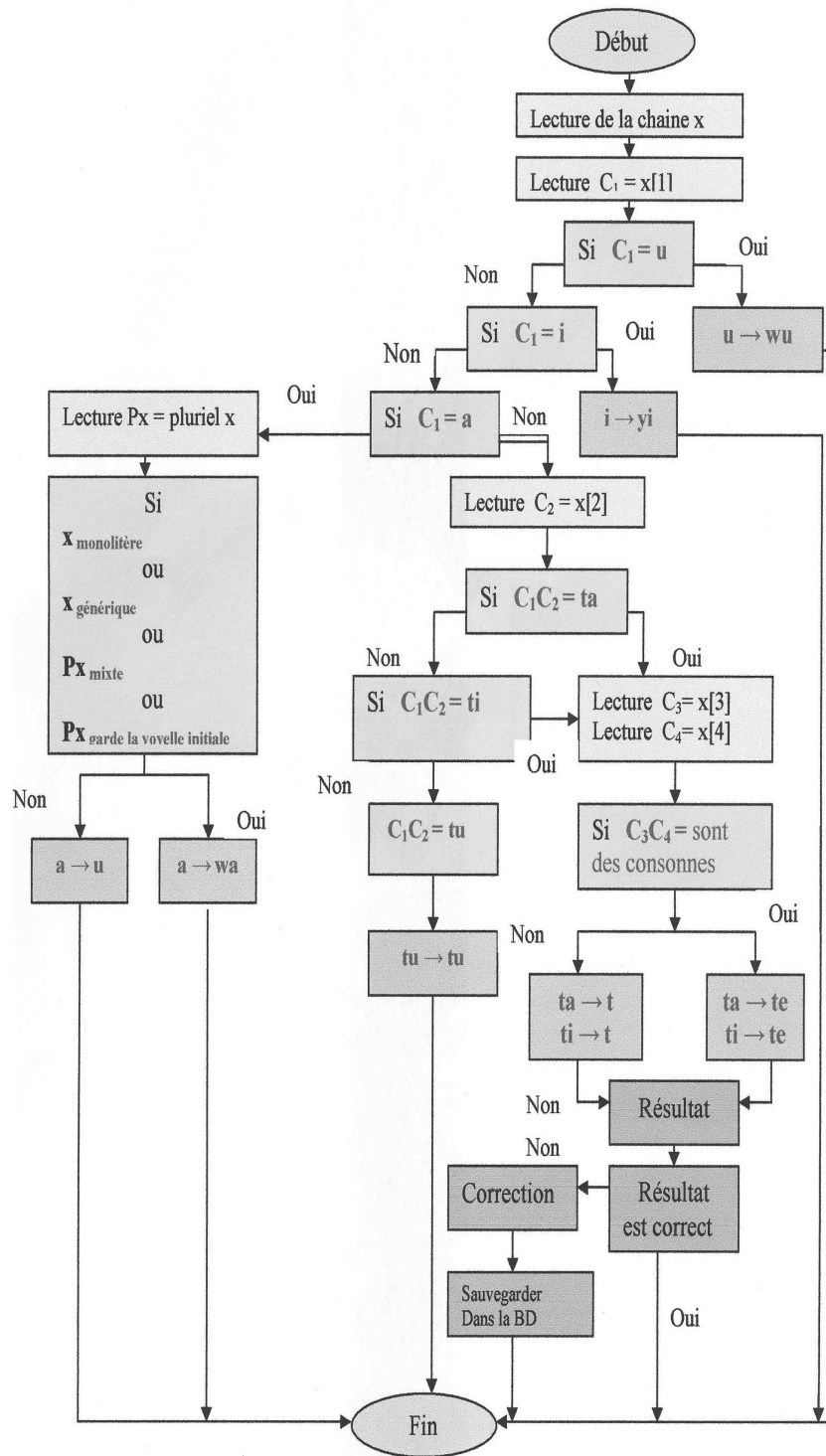
tu → tu

C : consonne

V : voyelle

3-3- Algorithme

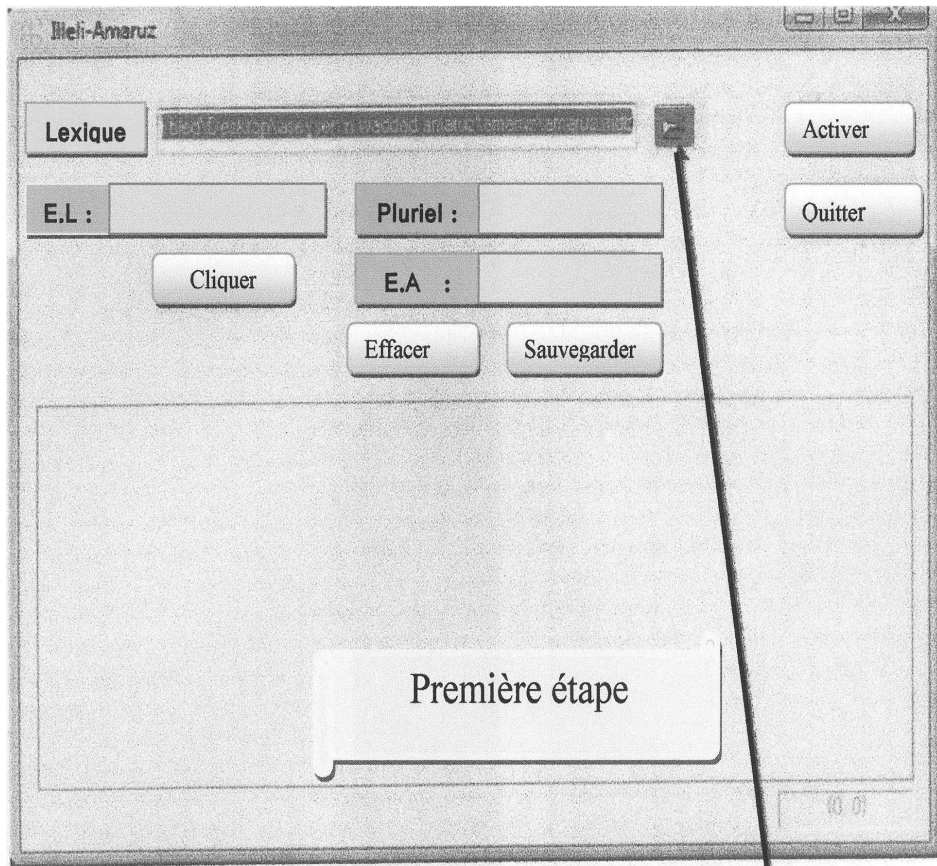
Pour concevoir notre logiciel, nous devons formaliser le problème et exprimer les actions à exécuter sous la forme d'un algorithme, pour ensuite le traduire dans un langage de programmation.



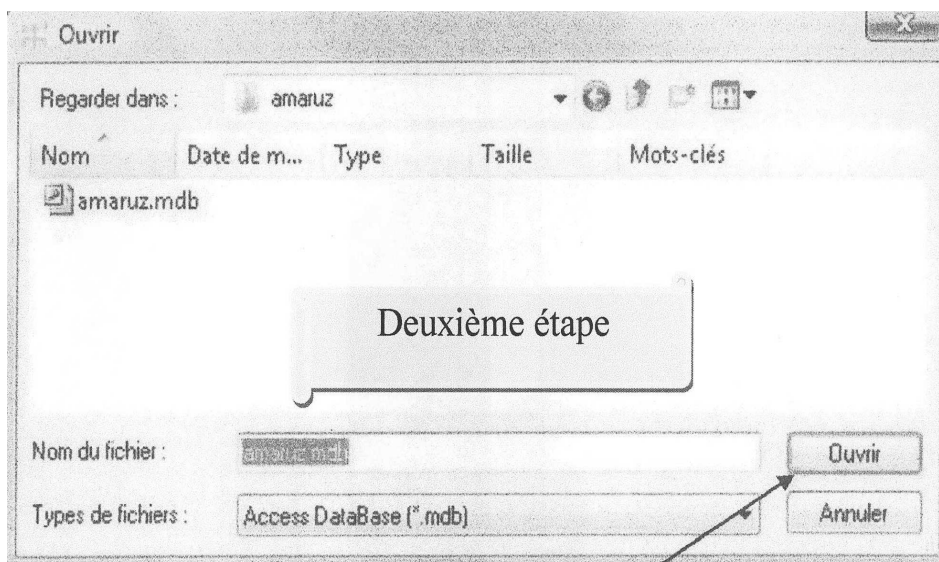
4- Application

E.L : état libre

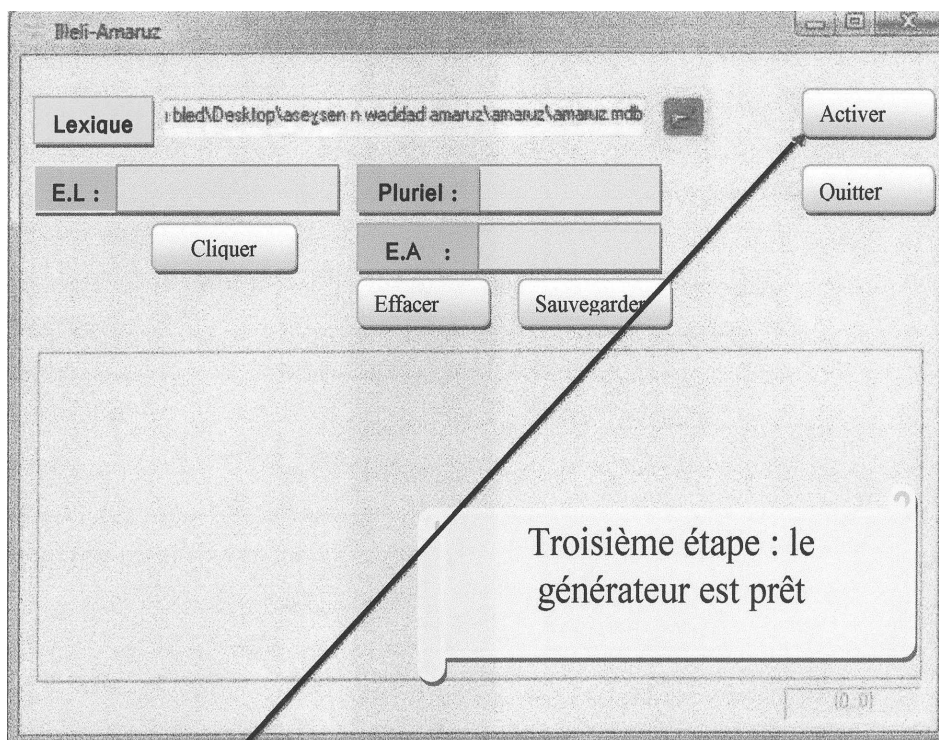
E.A : état d'annexion



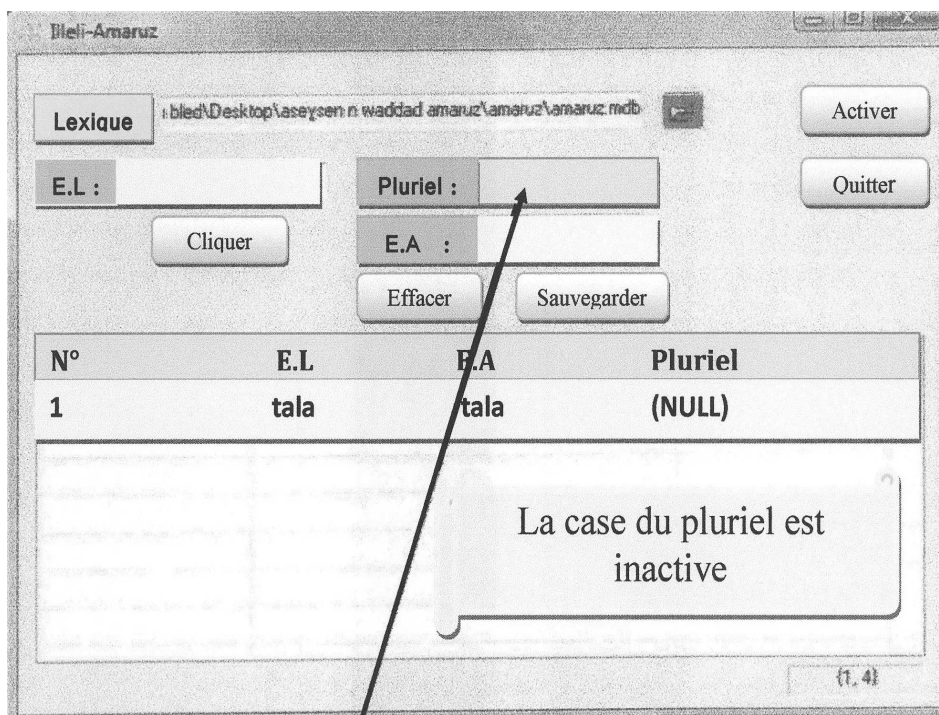
Cliquer pour entrer



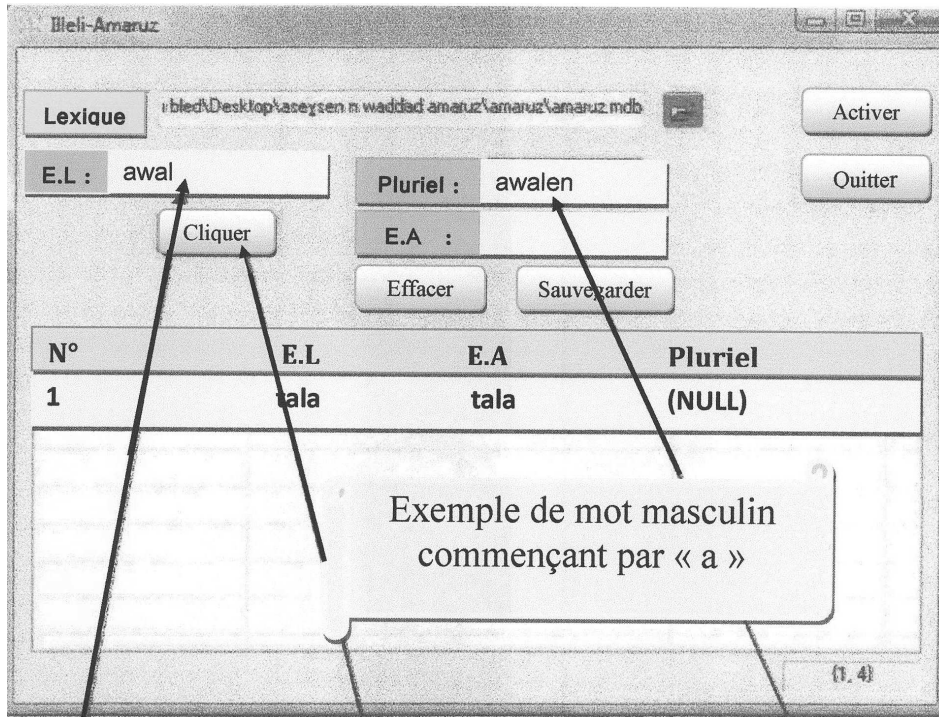
Cliquer sur 'Ouvrir' pour aller
à l'étape 3



Cliquer pour activer le
générateur



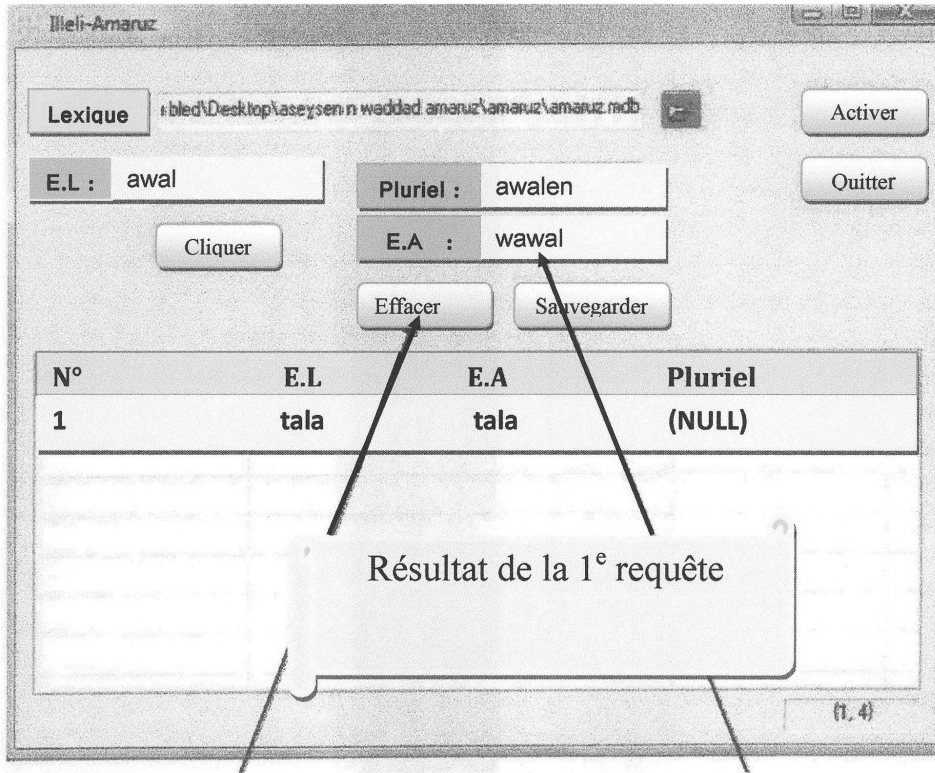
Cette case est active seulement quand on introduit un mot qui commence par « a » pour nous demander son pluriel.



1- On écrit un mot qui commence par « a »

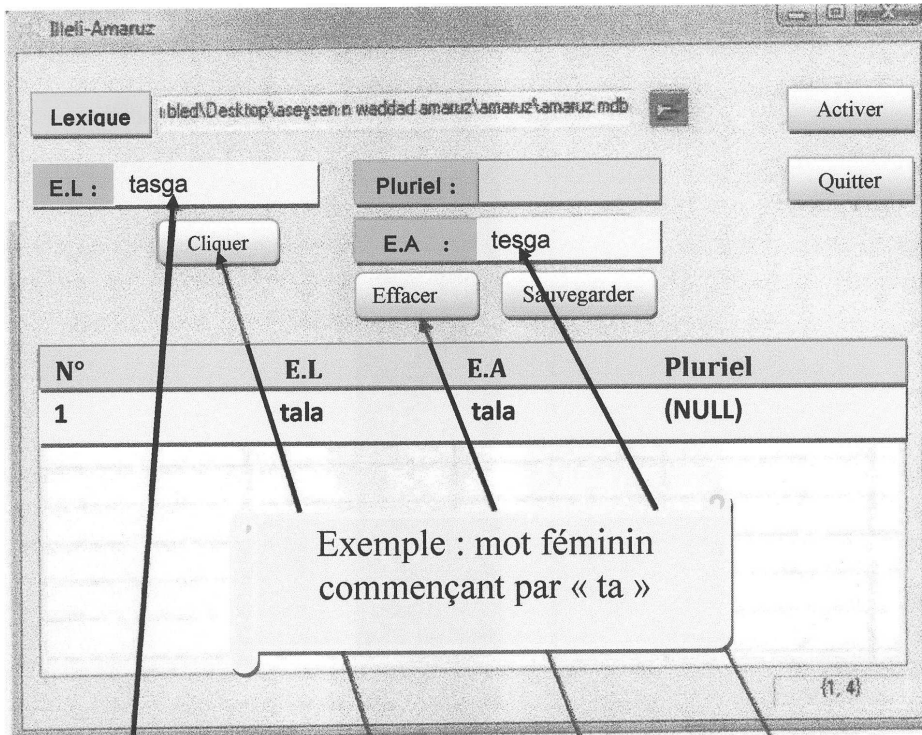
3- On clique sur ce bouton

2- Il nous demande d'écrire le pluriel du mot



2- Ce bouton permet de préparer le générateur à la prochaine opération en effaçant l'opération précédente

1- Résultat de l'opération précédente

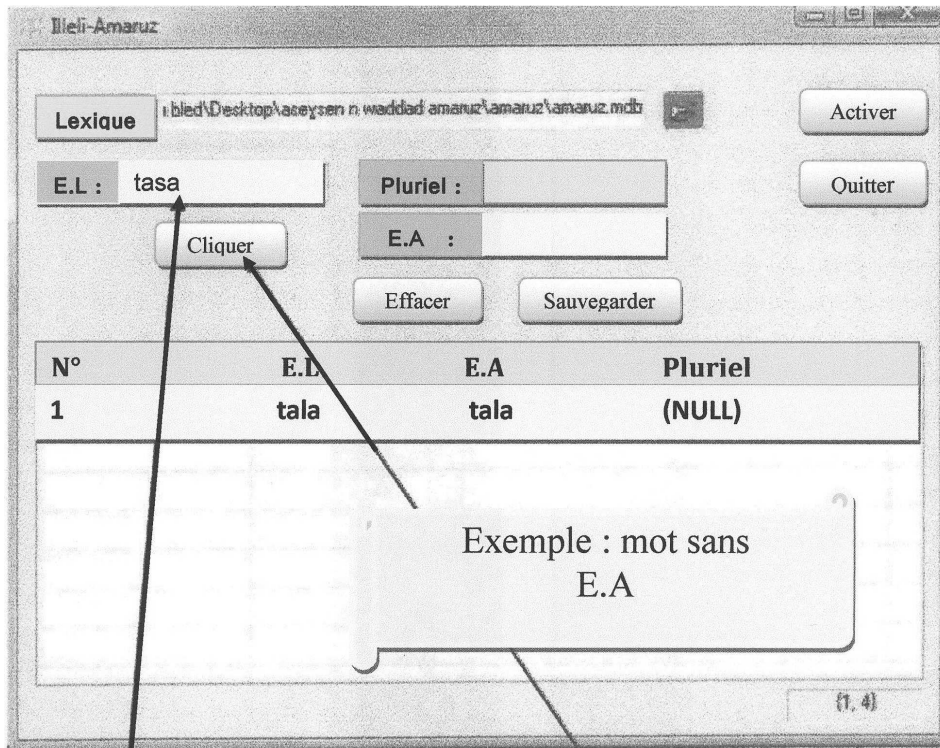


1- On introduit un autre mot qui commence par « t ». La case du pluriel est inactive

2- On clique sur ce bouton pour nous fournir son état d'annexion

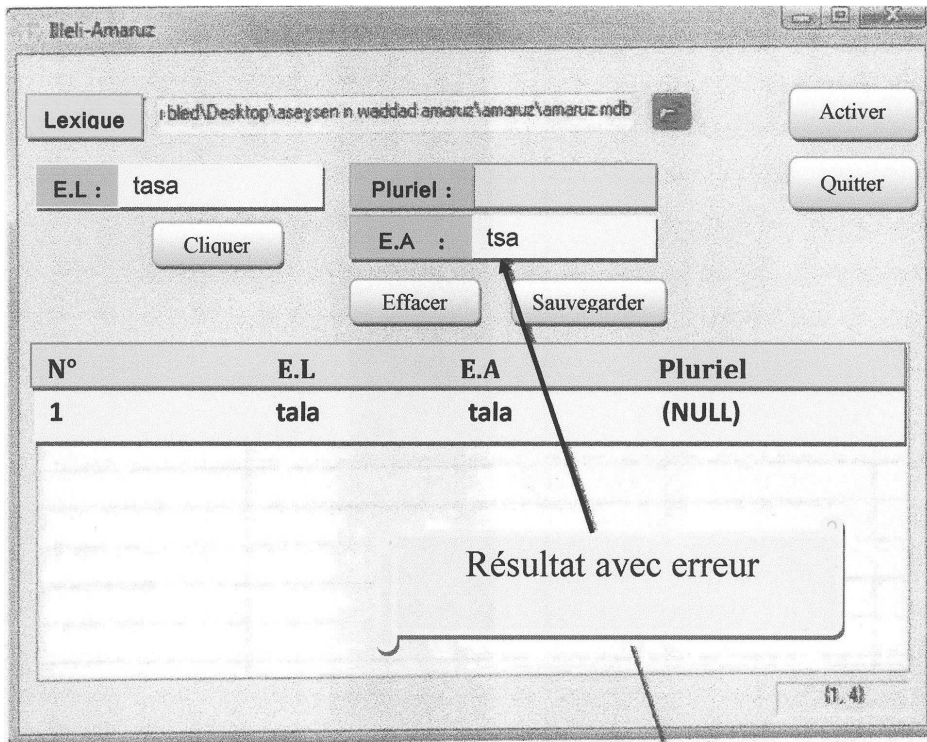
4- Effacer pour entamer une opération nouvelle

3- Résultat de l'opération

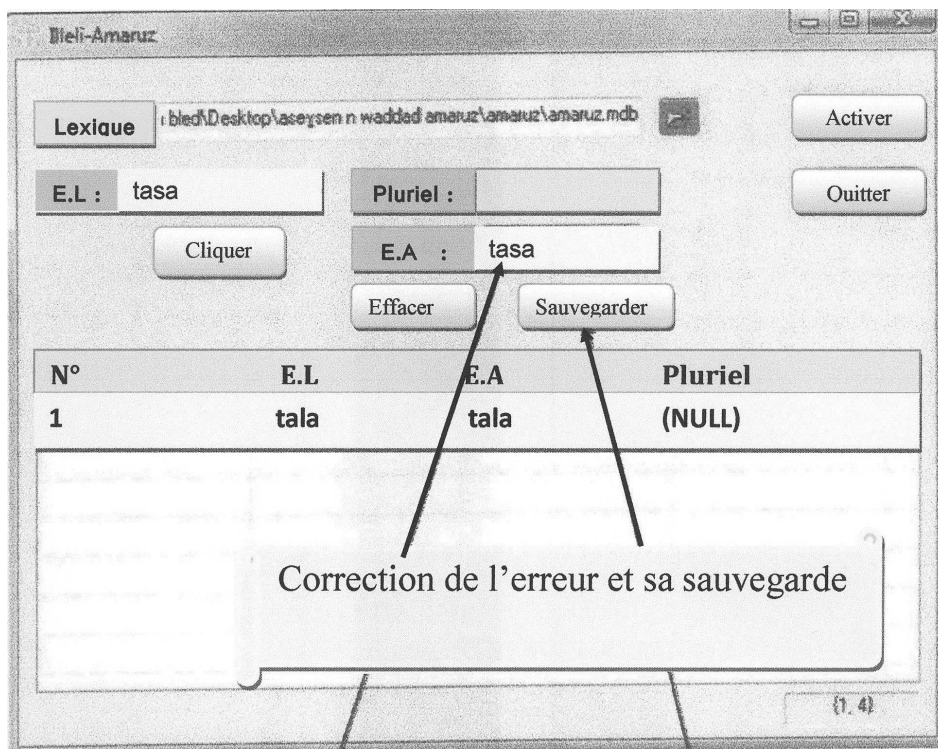


1- On introduit un mot monolitère féminin commençant par « ta »

2 - Ensuite on clique sur ce bouton

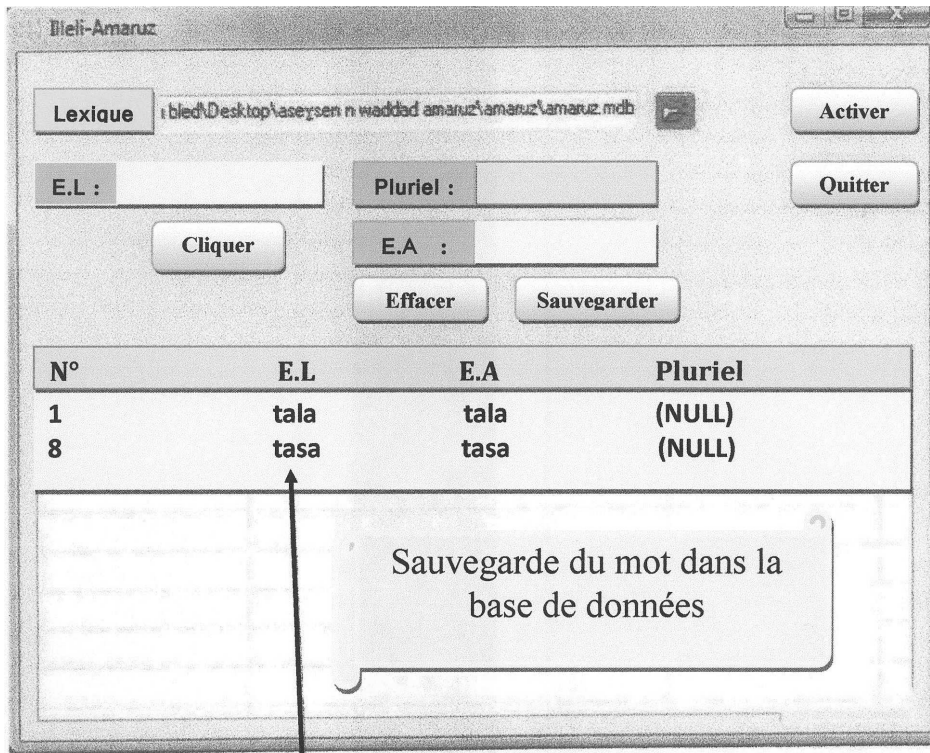


Le résultat de l'opération est faux, dans ce cas, on procède à la correction.

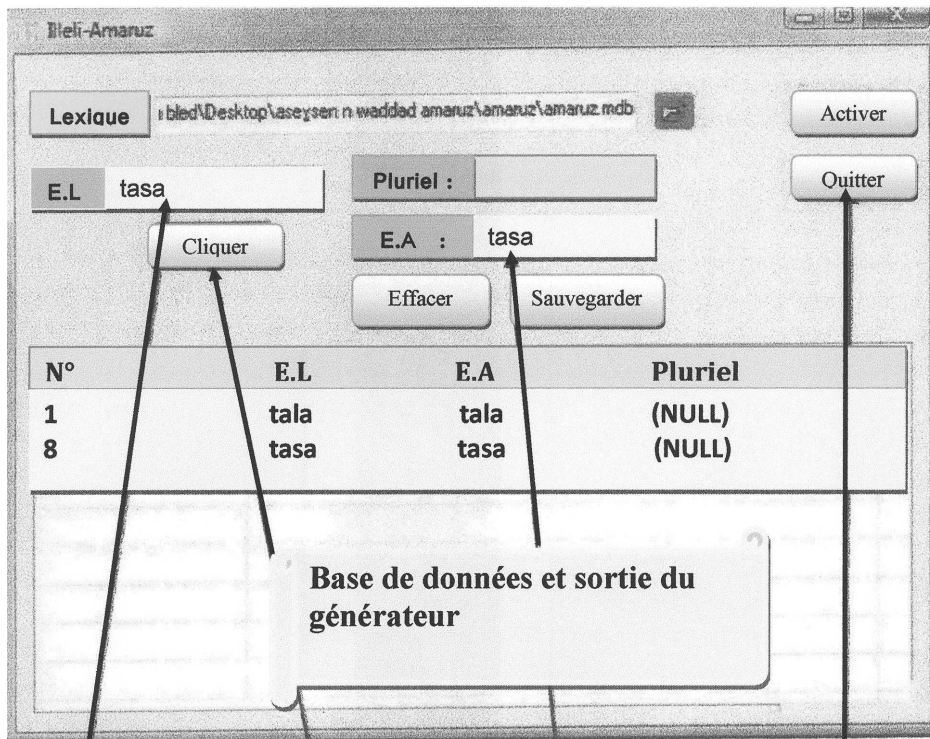


1- On corrige en écrivant « tasa » dans cette case

2- On clique sur ce bouton pour sauvegarder la réponse dans la base de données.



Le mot est enregistré dans la base de données, il apparaîtra à chaque fois qu'on le demande

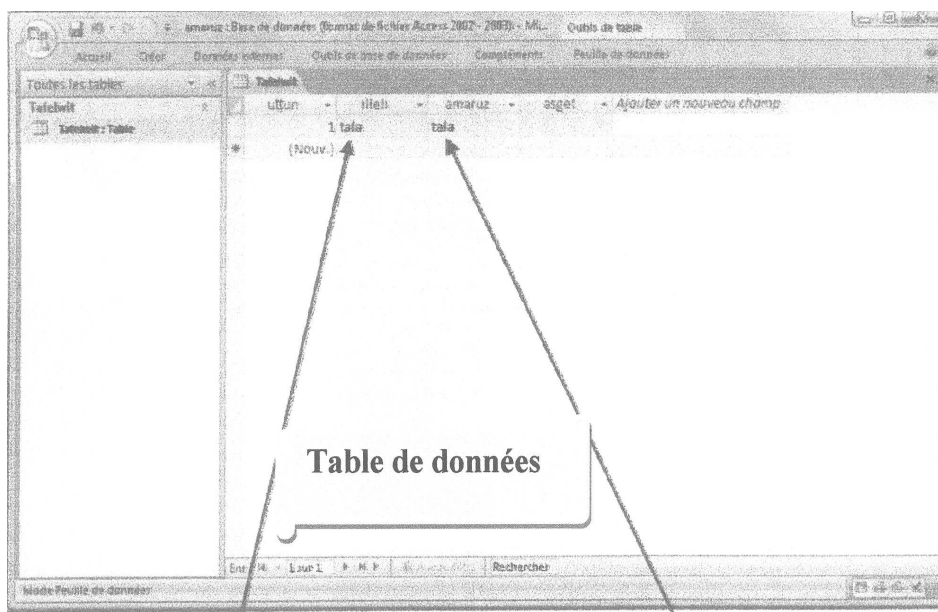


1- On propose le mot « tasa » pour la 2^e fois

2- On clique sur ce bouton

3- Le résultat est correct, puisqu'il existe dans la base de données

4- On clique sur ce bouton pour sortir



Zone de texte dans laquelle on peut écrire le mot à l'état libre

Zone de texte dans laquelle on écrit le mot à l'état d'annexion

Conclusion

Ce travail qui porte sur la génération automatique de l'état d'annexion du nom amazighe et, qui avait fait l'objet d'une étude par des étudiants du département de langue et culture amazighes de Béjaïa, en vue de l'obtention du diplôme de licence, sous la direction de M. Mahrazi. est sans doute l'un des très rares travaux réalisés dans ce domaine, car le traitement automatique de la langue amazighe n'est qu'à sa phase embryonnaire. Son objectif est double : d'une part, il peut servir dans l'enseignement assisté par ordinateur de la langue amazighe (EAO), en particulier l'état d'annexion, et d'autre part, il servira comme base de données lexicales où tous les noms dont l'état d'annexion fait exception y figure. Ce travail ouvre des perspectives à d'autres travaux qui porteront sur le traitement automatique de la langue amazighe, notamment le *correcteur orthographique*.

Par ce modeste travail, nous espérons que nous avons apporté une contribution, aussi minime soit-elle, à la standardisation de la langue amazighe par le biais du traitement automatique des langues naturelles.

Bibliographie

- BASSET A., 1932 : « Note sur l'état d'annexion en berbère », Bulletin de la Société de Linguistique de Paris, 33/2, p. 173-174.
- BEAULIEU F. 2004 : *Introduction à la programmation Orientée Objet, Pascal-Delphi*. Ed. Berti, Alger.
- BELAID M.C., 2007 : *Le champion de la programmation Delphi 5,6 et 7*. Ed. Pages Bleues Internationales, Alger.
- BELKHEMSA H. & D. CHAMEK 1999 : « Réalisation d'un analyseur lexico-syntaxique d'un corpus de la langue berbère », Mémoire d'ingénieur d'Etat, Université de Tizi-Ouzou.
- BENDAHMANE N., 2008 : *Créer une base de données*. Ed. l'Abeille, Tizi-Ouzou.
- BOUILLON P. et al., 1998 : *Traitements automatiques des langues naturelles*. Ed. DUCULOT.
- BOUKHRIS F. et al. 2008 : *La nouvelle grammaire de l'amazighe*. IRCAM, Rabat.
- BRUGNATELLI V., 1987 : « Deux notes sur l'état d'annexion en berbère », Proceedings 4th International Hamito-Semitic Congress, p. 349-359.
- CARRE R. et al. 1991 : *Langage humain et machine*. Paris, Presses du CNRS.
- CHAKER S., 1996 : *Manuel de linguistique berbère –II Syntaxe et diachronie*. Ed. ENAG, Alger.
- CHOUYA F. & R. ACHOUI & O. AINOUCHE, 2010/2011 : « Asili n umsirew n waddad amaruz ». Mémoire de licence, sous la direction de M. Mahrazi. Université de Béjaïa.
- DALLET J.-M., 1982 : *Dictionnaire Kabyle-Français*. Ed. SELAF, Paris.
- FLEURY M., Sans date : *Enseignement assisté par ordinateur*. Université Laval.
- FUCHS C. et al., 1993 : *Linguistique et traitements automatiques des langues*. Paris, Hachette.

- GARDENT Cl. et K. BASHUNG 1994 : *Techniques d'analyse et de génération pour la langue naturelle*. Clermont-Ferrand, Adosa.
- GARCIA B., Sans date : *Introduction à C++ Builder* OURS BLANC DES CARPATHES ISIMA 1988-1999.
- GRAÏNE S., 2009: *Delphi : Concepts de base, bases de données*. Ed. l'Abeille, Tizi-Ouzou.
- HAMEK B. et al., 2055 : *Ilugan n tira*. Editions TALANTIKIT.
- IMARAZEN M., 2007 : *Manuel de syntaxe berbère*, H.C.A, Alger.
- NAIT-ZERRAD K., 1995 : *Grammaire du berbère contemporain (kabyle), I – Morphologie*, ENAG, Alger.
- NAIT-ZERRAD K., 1996 : *Grammaire du berbère contemporain (kabyle) : II- Syntaxe*, Ed. ENAG. Alger.
- NAIT-ZERRAD K., 2004 : *Linguistique berbère et Application*, L'Harmattan. Paris.
- SADIQI F., 1997 : *Grammaire du berbère*, Editions l'Harmattan.
- OULD AMROUCHE, 1999 : « Réalisation d'un logiciel de traitement de données lexicales berbères », Mémoire d'ingénieur d'Etat, Université de Tizi-Ouzou.
- PIERREL J-M., 2000 : *Ingénierie des langues*. Hermès, Paris.
- SABAH G., 1988 : *L'intelligence artificielle et le langage : représentation des connaissances*. Volumes 1et 2, Paris Hermès.
- TAREK A. & N. OUDAHMANE, 1999 : « Contribution au traitement automatique de la langue amazighe », Mémoire d'ingénieur d'Etat, Université de Tizi-Ouzou.
- ZOCK M. et G. SABAH, 1992, "La génération automatique de textes : trente ans déjà ou presque", in *Langages*, 106, pp. 8-35.

1-HAMEK B. et al., 2055 : *Ilugan n tira*. Editions TALANTIKIT, pp. 24-26.