L'Impact Informatique de l'Intégration de la Langue Arabe dans les Téléphones Mobiles

A. ABDELHADI[1], M IBN MOHAMMED[2] O. KADRI[3]
Laboratoire d'automatique et de productique, Université de Batna, Algérie 1,3
Département d'informatique, Université de Constantine, Algérie2
[1]abdelhadi.adel@yahoo.fr
[2]ibnm@yahoo.fr
[3]ouahabk@yahoo.fr

Résumé: Le travail présenté entre dans le cadre du domaine d'interface homme machine. Notre objectif est d'étudier l'impact informatique de l'intégration de la langue arabe dans les téléphones mobiles, afin de réaliser une interface homme machine arabe. L'affichage correct des caractères arabes est indispensable dans une interface graphique. Puisque les caractères arabes changent leurs formes selon la position qu'ils occupent dans un mot, alors il est nécessaire de faire une analyse contextuelle sur chaque mot, pour trouver la forme correcte de chaque caractère.
La transformation de deux ou plusieurs caractères en une seule forme, demande un traitement particulier, comme dans le cas de la ligature LAM-ALEF arabe.
La langue arabe possède une direction d'écriture différente par rapport aux autres langues embarquées dans les téléphones mobiles, ce qui exige de trouver un algorithme bidirectionnel qui assure un affichage correct des messages SMS. Ces messages peuvent contenir des caractères de direction différente, de droite à gauche, de gauche à droite ou des caractères qui n'ont pas de direction. Il permet de rendre l'affichage des messages compréhensible.

Mots clés: Caractère Arabe, Analyse Contextuelle, Glyphe, Ligature, Algorithme Bidirectionnel, Ordre Logique, Ordre Visuel.

Introduction

La langue arabe possède plusieurs caractéristiques qui demandent des traitements particuliers pour qu'elle soit implémentée dans un programme, intégrée dans un PC ou dans un téléphone mobile. D'abord, la direction d'écriture est de droite à gauche. En plus, les caractères arabes changent leurs formes selon la position qu'ils occupent dans un mot. Il y a aussi la possibilité de transformer deux caractères en une seule forme. Cet article étudie les grands impacts pour intégrer une IHM (*Interface Homme Machine*) arabe dans les téléphones mobiles

L’interface graphique d’un téléphone mobile est composée d’un ensemble de boîtes de dialogues appelés : *écrans*. Un écran contient une ou plusieurs chaînes de caractères appelés *prompts***.** Dans le cas d’une MMI arabe, Les prompts sont constitues d’une chaîne de caractères arabes.

L’impact de la langue arabe sur les téléphones mobiles apparaît surtout au niveau de l’affichage des caractères arabes. Puisque les messages arabes s’écrivent de droite à gauche alors que le code des caractères est stocké en mémoire de gauche à droite.

Une analyse contextuelle sur les prompts arabes est nécessaire pour qu’ils soient affichés correctement, puisque, selon la position du caractère arabe dans un mot, il peut avoir une forme différente soit ; au début, au milieu, à la fin ou isolé. La transformation de deux caractères en une seule forme est possible dans la langue arabe. En revanche, cette transformation a besoin de faire un traitement particulier pour assurer l’affichage correct de cette forme.

L’IHM des téléphones mobiles propose des menus, des écrans, mais surtout la possibilité de l’édition et l’affichage des messages courts qui s’appellent : SMS (*Short Message Service*).Un message SMS est composé d’une chaîne de caractère appartenait à une seule langue ou plusieurs. L’édition et l’affichage correct d’un message SMS bilingue a besoin d’un algorithme d’affichage bidirectionnel.

L'informatisation de la langue arabe

L'arabisation des systèmes informatiques monopostes ou Multi-Utilisateurs concerne en effet, les deux aspects fondamentaux des composantes matérielles d'une part et des programmes incluant les langages de programmation, les utilitaires, et les systèmes opératoires de l'autre.

L'arabisation est orientée vers l'accomplissement des objectifs suivants :

Rendre l'usage de la langue arabe en informatique aussi simple et efficace que les langues latines,

Concevoir des applications indépendantes de la langue de dialogue,

Gérer des données en plusieurs langues simultanément,

Concevoir des applications intégralement indépendantes de la langue de dialogue choisie par l'utilisateur et intégrant simultanément les différentes langues.

Parmi ces problèmes, certains ont leur origine dans la langue arabe elle-même, d'autres sont issues de la nécessité de mixer la langue arabe avec d'autres langues d'origine latines, d'autres proviennent directement des champs d'application des normes et des codes de représentation [BEN 03].

1.1. Codage informatique de l’alphabet arabe

Le standard de codage ASCII a montré son insuffisance, puisqu’il fonctionne sur 8 bits, c’est à dire qu’il ne permet que 128 positions de codage. Actuellement, il y a plus d’un million de caractères dans le monde entier qui ont besoin de codage, pour satisfaire les demandes croissantes des langues industrielles non anglo-saxonnes, et pour permettre à d'autres langues comme l’arabe et ses règles de liaisons d'apparaître sur un écran informatique.

Cette insuffisance a obligé les constructeurs d’ordinateurs de créer un autre standard de codage, qui peut supporter ce nombre énorme de caractère, mais compatible avec les normes existantes, c’est la norme Unicode ou bien UCS. Ce standard a été créé par un groupe de constructeurs d’ordinateurs en 1989. Il permet de définir le codage pour la majorité des caractères utilisés par les langues du monde. C’est un jeu de codage sur deux ou plusieurs octets, et pour assurer la compatibilité avec la norme d’ASCII, les 256 premiers caractères sont réservés pour L’ISO-Latin-1[FAN 99].

Arabisation des téléphones mobiles

La base de l’arabisation d'un système consiste en l’adaptation de tous les aspects concernant le système d'exploitation pour permettre à l'utilisateur de gérer le texte arabe ou niveau d'entrée ou bien de la sortie.

Les spécifications avancées de l’arabisation dans le domaine des téléphones mobiles peuvent inclure beaucoup d'autres particularités comme l'entrée prédictive (comme un outil de travail à la limitation du clavier numérique du téléphone)

La base de l’arabisation demande de répondre à deux questions principales suivantes :

- **Au niveau d’entrée** : comment l'utilisateur réagit avec le système pour entrer un texte arabe ?
- **Au niveau de sortie** : comment le système affichera les données arabes qui ont été entrés par l'utilisateur?

Et certains d'autres problèmes comme :

- **L’alignement de Texte** : accepter et gérer un texte aligné à gauche (latin) ou juste aligné à droite (arabe).
- **Gestion de Curseur** : qui est utilisé pour traiter les positions de curseur, soit avant soit après un caractère dans un texte bidirectionnel.

Les exigences demandées pour traiter toutes ces spécifications sont :

- ✓ Un Traitement bidirectionnel,
- ✓ Un changement des glyphes,

- ✓ Une gestion des voyelles (**CHACL**),
- ✓ Des Fontes arabes d'une taille supportée par le système.

La figure 1 permet de définir un système de téléphone mobile:

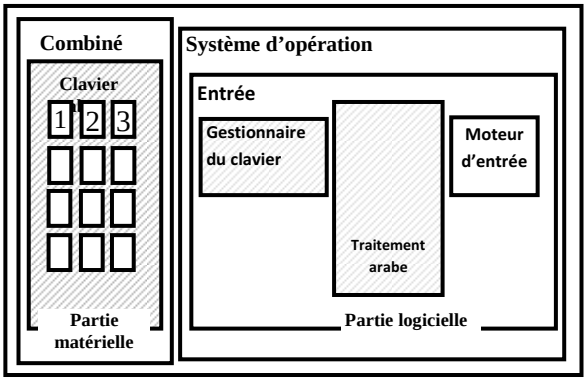

Figure 1: Système de téléphone mobile

La Figure 2 montre tous les modules pour construire un système de téléphone mobile ainsi que tous leurs connexions. Le scénario typique, décrivait ci-dessous, où tous les modules du système sont impliqués, montre toutes les étapes d'affichage d'un message arabe.

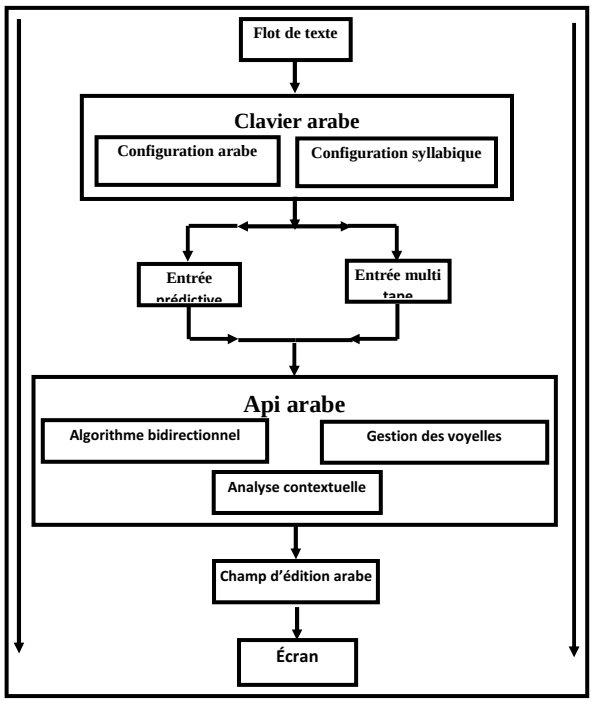

Figure 2: Description intérieure du système de téléphone mobile

Algorithme d'Analyse Contextuelle

Cette partie a pour but d'expliquer les différentes étapes du développement de l'algorithme d'analyse contextuelle (**GLYPH SHAPING)**. Cet algorithme permet de faire le changement des glyphes de chaque caractère arabe en fonction de sa position dans un mot pour qu'il apparaisse de manière correcte lorsqu'il est affiché sur l'écran du téléphone mobile.

Notre algorithme est utilisé dans le cadre d'un algorithme plus général d'affichage de texte bidirectionnel dans un message SMS pour les téléphones mobiles.

1.2. Principe de l'algorithme

L'analyse contextuelle est nécessaire pour plusieurs langues d'écriture afin de présenter leurs caractères avec une forme correcte. Pour la langue arabe, afin de choisir la forme de présentation appropriée d'un caractère, il est nécessaire de prendre en considération la forme de leurs caractères voisins s'ils existent.

L'algorithme de **GLYPH SHAPING** est basé sur quatre tableaux représentant les différents types de glyphes que peut prendre un caractère arabe.

1.3. L'automate de l'algorithme d'analyse contextuelle

Dans l'arabe, les lettres d'une série changent progressivement leurs formes avec l'utilisation de l'analyseur contextuel. En effet, seulement les deux dernières lettres de cette série changent. Nous utilisons des automates d'états finis pour mettre en œuvre notre algorithme. Les états sont les formes des deux dernières lettres, et les transitions sont étiquetées par le type du caractère précédent. Le caractère précédent indique celui qui précède le dernier caractère lu par l'analyseur. Seulement deux actions sont possibles : " le caractère de la classe (**R**)" ce qui signifie que le caractère précédent appartient au tableau des caractères de la classe R, et le caractère de la classe (**D**)" ce qui signifie le contraire.

La figure 8 présente le graphe associé à notre automate d'analyse contextuelle :

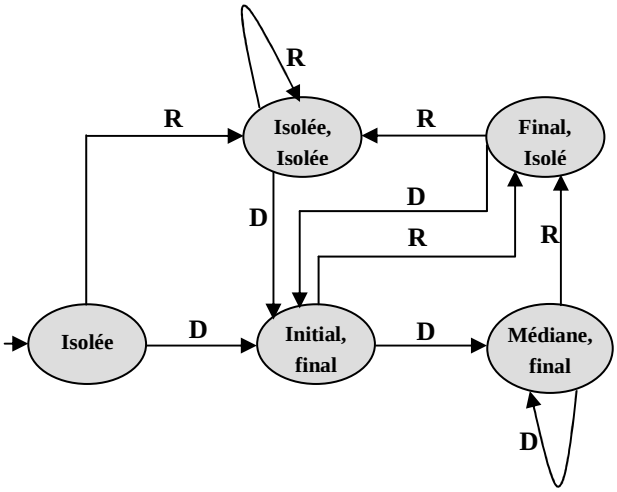

Figure 3: Graphe d'automate d'analyse contextuel

L'analyseur contextuel reçoit un caractère ayant la forme isolée. Donc l'état initial est composé seulement de la première forme de lettre. Il y a une transition d'un état à un autre après la lecture d'un nouveau caractère. L'automate s'arrête quand la série finit. Ainsi, tous les états peuvent être vus comme des états finals.

La figure suivante montre un exemple d'utilisation de cet automate d'état fini :

Analyser le mot : نظام

- **Lire le caractère = ن**
  Analyser = ن ("isolée" état = l'état initial)

- **Lire le caractère = ظ**
  Le caractère précédent (ن) est de la classe D

  La transition "D" est sélectionnée

  Analyser = نظ (l'état "initial, final")

- **Lire le caractère = ا**
  Le caractère précédent (ظ) est de la classe D

  La transition "D" est sélectionnée

  Analyser = نظا (l'état "médiane, final")

Tab. 1: Exemple de l'analyseur contextuel

Algorithme d'Affichage Bidirectionnel

Cette partie a pour but d'expliquer les différentes étapes de l'Algorithme d'Affichage Bidirectionnel (**BIDI MOBILE**). Cet algorithme permet de préparer une chaîne de caractères pour l'afficher sur l'écran du téléphone mobile, qui s'écrit dans deux directions différentes : de droite à gauche (comme l'Arabe), de gauche à droite (comme les langues latines) et qui permet aussi de résoudre le problème des caractères qui n'ont pas de direction (caractères neutres).

Cet algorithme est utilisé dans le cadre de l'intégration de l'Arabe dans les téléphones mobiles. Elle sert à préparer les chaînes de caractères constants (prompts) d'une **MMI** et permet d'assurer aussi l'édition et l'affichage correct des messages **SMS**.

1.4. Principe de l'algorithme

Avant de commencer l'algorithme d'affichage bidirectionnel (**BIDI MOBILE**), la première chose à faire est de traiter la liaison qui se trouve entre les caractères arabes, en appliquant l'algorithme d'analyse contextuel (**GLYPH SHAPING).**

Cet algorithme permet de remplacer les caractères arabes par leurs glyphes corrects.

Il y a une autre tâche indépendante de l'algorithme bidirectionnel ; c'est l'opération de découpage du texte traité en ligne. La solution proposée est basée sur la propriété

de directionalité de chaque caractère.

Cette solution est divisée en trois parties essentielles :

✓ **Définition de la directionalité d'affichage** : cette partie est chargée de définir le type directionnel de chaque caractère et de préciser la base de direction d'affichage.

✓ **Direction d'affichage droite à gauche** : si le type du premier caractère de la chaîne d'entrée est de type **R,** alors la direction de toute la chaîne de sortie est de droite à gauche.

✓ **Direction d'affichage gauche à droite :** si le type directionnel du premier caractère de la chaîne d'entrée est de type **L,** alors toute la chaîne de sortie prend la direction gauche à droite.

1.4.1. *Définition de la directionalité d'affichage*

a. *Définition du type directionnel des caractères*

La définition du type de direction de chaque caractère de la chaîne d'entrée est indispensable pour l'affichage correct du message. Le codage proposé est le suivant :

R	**Caractère arabe et hébreu**	**0**
L	**La plupart des caractères alphabétiques**	**1**
AN/ EN	**Chiffre**	**2**
N	**Caractère neutre**	**3**

Tab. 2: Définition du type directionnel de chaque caractère

b. *Définition de la base de direction d'affichage*

Dans cette partie, on fait la définition d'une base de direction d'affichage pour la chaîne d'entrée.

Ce traitement est basé sur le type du premier caractère de forte direction rencontrée et permet de distinguer ces deux cas suivants :

- Direction d'affichage droite à gauche.
- Direction d'affichage gauche à droite.

1.4.2. *Direction d'affichage droite à gauche*

D'abord, la direction de toute la chaîne d'entrée est de droite à gauche. La solution proposée est basée sur le fait de trouver toute la chaîne successive des caractères de type **R** et les caractères de type **N.**

Dans ce cas là, il y a plusieurs règles de résolutions des caractères neutres à appliquer pour donner une direction à chaque caractère **N**.

Dans le cas où il y aurait une ambiguïté pour définir la direction d'un caractère **N**, comme il est indiqué dans les deux règles suivantes :

R	**N**	**eor**	**R**	**i**	**eor**
sor	**N**	**L**	**sor**	**i**	**L**

Tab. 8: *Direction du caractère N*

Le caractère en question **(i)** prend dans ce cas la direction de la chaîne d'entrée courante.

Ensuite, il faut trouver la manière correcte d'affichage de ces caractères. La solution est d'inverser la chaîne trouvée d'une manière complète, et de la mettre à la fin de la chaîne de sortie.

Dans tous les exemples qui vont être présentés :

XX : les caractères majuscules représentent des caractères arabes.

xx : les caractères minuscules représentent des caractères latins.

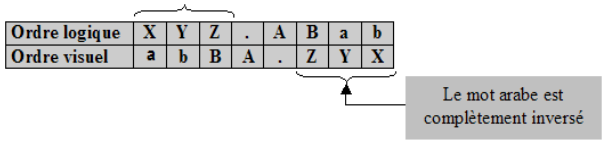

Figure 4: Inversement de chaque mot arabe

La recherche de la chaîne de caractères qui sera affichée de droite à gauche, peut contenir des caractères de type **R** et de types **N**.

Pour qu'un caractère de type **N** appartienne à la chaîne de caractère **R,** il faut qu'il vérifie certaines conditions, de telle sorte qu'ils ne soient pas entre :

- Deux caractères de type **L,**
- Deux chiffres,
- L'une des deux bornes soit un caractère de type **L** et l'autre borne soit un chiffre ou l'inverse.

Le traitement appliqué sur une suite de chiffres ou bien de caractères **L** est le même. Premièrement fixer son emplacement, c'est à dire définir le début et la fin dans la chaîne. Deuxièmement, la réorganisation de cette chaîne se fait de la manière suivante :

La chaîne prend le même ordre que dans la chaîne d'entrée, mais avec un certain décalage, selon la présence ou l'absence d'une chaîne de caractère **R** au début.

Exemple1	Ordre logique	X	Y	a	b	c	1	2	
	Ordre visuel		a	b	c	1	2	Y	X
Exemple 2	Ordre logique	A	B	1	2	3	4	5	6
	Ordre visuel	1	2	3	4	5	6	B	A

Tab. 3: direction d'affichage droite à gauche

D'après les exemples précédents, nous constatons que les mots de direction R sont complètement inversés, par contre les mots de direction L et les chiffres gardent le même ordre avec un certain décalage.

1.4.3. *Direction d'affichage gauche à droite*

Cette partie permet de faire la recherche d'une chaîne des caractères **L**, chiffres ou neutres.

Les caractères neutres doivent vérifier certaines conditions de telle sorte qu'ils soient entre :

- Deux caractères de type **L**,
- Deux chiffres,
- L'une des deux bornes soit un caractère de type **L** et l'autre borne soit un chiffre, soit l'inverse.

Cas 1	Ordre logique	a	b		c	d	X	Y	
	Ordre visuel	a	b		c	d	Y	X	
Cas 2	Ordre logique	a	b	C	D	1	3	,	4
	Ordre visuel	a	b	D	C	1	3	,	4
Cas 3	Ordre logique	a	b	1		c	d	Y	Z
	Ordre visuel	a	b	1		c	d	Z	Y

Tab. 4: direction d'affichage gauche à droite

Le processus de réorganisation de la chaîne de caractères est réalisé de la manière suivante: Une suite de caractères **L** et neutres reste dans le même ordre sans décalage. Par contre, les chaînes de caractères **R** sont inversées et mettent à côté de la chaîne de caractère **L**.

Résultats obtenus

Voici plusieurs tableaux qui montrent une comparaison entre les résultats obtenus par BIDI MOBILE et les résultats fournis par la référence d'Unicode.

On peut constater trois cas possible :

- Direction d'affichage droite à gauche.
- Direction d'affichage gauche à droite.
- Direction d'affichage neutre

XX : les caractères majuscules représentent des caractères de direction R.
xx : les caractères minuscules représentent les caractères de direction L.
N : le reste représente les caractères neutres

Teste	La chaîne source	Référence D'Unicode	Résultats du BIDI MOBILE
1	CAR IS the car IN ENGLISH	HSILGNE NI the car SI RAC	HSILGNE NI the car SI RAC
2	NUMBER IS 123456	123456 SI REBMUN	123456 SI REBMUN
3	TEST 23 ONCE abc	abc ECNO 23 TSET	abc ECNO 23 TSET
4	TEST ABC 123456	123456 CBA TSET	123456 CBA TSET
5	SOLVE 1*5 1-5 1/5 1+5	1+5 1/5 1-5 5*1 EVLOS	5+1 5/1 5-1 5*1 EVLOS
6	HE SAID “it is 123, 456, ok”	“ok ,456 ,123 it is” DIAS EH	“ok ,456 ,it is 123” DAIS EH
7	HE SAID “it is a car!” AND RAN	NAR DNA “!it is a car” DIAS EH	NAR DNA “!it is a car” DIAS EH
8	HE SAID “it is a car!x” AND RAN	NAR DNA “it is a car!x” DIAS EH	NAR DNA “it is a car!x” DIAS EH
9	THE RANGE IS 2.5..5	5..2.5 SI EGNAR EHT	5..2.5 SI EGNAR EHT
11	CHANGE -10%	%10- EGNAHC	%10- EGNAHC
12	TEST ~~~23%%% ONCE abc	abc ECNO 23%%%~~~ TSET	abc ECNO %%%23~~~ TSET

Tableau 5 : Résultats obtenus avec la direction d’affichage droite à gauche

Teste	La chaîne source	Référence D'Unicode	Résultats du BIDI MOBILE
1	car is THE CAR in Arabic	car is RAC EHT in arabic	car is RAC EHT in arabic
2	number is 1234567	number is 1234567	number is 1234567
3	he said AB “IT IS OK 123”	he said “ 123 KO SI TI” AB	he said OK SI TI” BA 123 ”
4	he said “IT IS 123, 456, OK”	he said “KO ,456 ,123 SI TI”	he said “SI TI 123, 456, KO”
5	he said “IT IS (123, 456), OK”	he said “KO ,(456 ,123) SI TI”	He said “0K (456 ,123) IT IS“
6	he said “IT IS A CAR!X”	he said “X!RAC A SI TI”	he said “X!RAC A SI TI”
7	he said “IT IS A CAR!”	he said “RAC A SI TI!”	he said “RAC A SI TI!”
8	abc (TEST)	abc (TSET)	abc)TSET(
9	he said “A SI TI bmw KO ,500.”	he said “A SI TI bmw KO ,500.”	he said TI SI A “ bmw KO ,500.”
10	a/1	a/1	a/1

Tableau 6 : Résultats obtenus avec la direction d’affichage gauche à droite

Conclusion

L'objectif de notre travail était l'étude de l'impact informatique de l'intégration de la langue arabe dans les téléphones mobiles, afin d'arriver à réaliser une interface homme-machine à la portée du monde arabe. Dans cet article nous décrivons essentiellement la mise en œuvre de plusieurs solutions destinées à résoudre les problèmes posés par les caractéristiques de la langue arabe.

Nos travaux présentés dans cet article concernant l'explication des étapes suivies pour le développement de deux algorithmes, l'algorithme d'analyse contextuelle (Glyph Shaping) et l'algorithme d'affichage bidirectionnel (Bidi Mobile).

L'algorithme d'analyse contextuelle destinée essentiellement à traiter le problème de la liaison qui se trouve entre les caractères arabes et définir leur forme correcte, quel que soit le type des caractères voisins, arabes ou bien autres. Il faut noter que chaque caractère peut prendre plusieurs formes. Cet algorithme permet aussi de résoudre le problème de la ligature arabe LAM-ALEF, et ça pendant l'analyse de tous les caractères. Cet algorithme est utilisé par un autre algorithme, c'est celui de l'affichage bidirectionnel.

L'algorithme d'affichage bidirectionnel est un algorithme inspiré d'une solution complète, proposé par le standard Unicode destiné à traiter les complications de l'écriture de la langue arabe. Cette solution qui vise essentiellement le domaine des téléphones mobiles, et plus particulièrement, trouver une solution pour l'affichage correct des messages SMS.

Nous pensons que les objectifs visés par ce travail ont été atteints. De plus, les solutions proposées pour traiter les caractéristiques exigées par la langue arabe sont efficaces, faciles et à la portée des utilisateurs qui exploitent les téléphones mobiles intégrant ces solutions.

Cependant, ces résultats satisfaisants n'excluent pas certains compléments qui peuvent être apportés à notre travail. Il s'agit en particulier les points suivants :

Le traitement des voyelles associées aux caractères arabes qui sont généralement utilisées dans les textes religieux et pédagogiques. En revanche, l'utilisation de ces voyelles dans les messages SMS peut enlever l'ambiguïté entre certains mots.

L'amélioration de l'algorithme d'affichage bidirectionnel, afin de traiter les messages SMS bilingues compliqués, qui contiennent des caractères neutres, et possèdent la propriété miroir.

Le traitement des messages SMS représentant des opérations arithmétiques écrit en arabe, de droite à gauche.

Prendre en considération l'utilisation des dictionnaires des mots arabes, et cela au niveau de l'édition des messages SMS, en utilisant l'entrée prédictive des caractères.

Références

[1] S. Atkin and R. Stansifer Implementations of Bidirectional Reordering Algorithms, 18th International Unicode Conference Hong Kong, April 2004.

[2] M. Benhenda, vers une normalisation des pratiques de communication dans le contexte d'un multilinguisme intégral (arabe latin), Institut Supérieur de Documentation, Tunisie, 2003.

[3] M. Davis, the Bidirectional Algorithm, Unicode Standard Annexe9, mars 2005.

[4] A. Dean. "Optimized Implementations of Bi-directional Text Layout and Bi-directional Caret Movement." Thirteenth International Unicode Conference, September 1999.

[5] M. Fanton, TEX : les limites du multilinguisme, centre d'étude et de recherche en traitement automatique de l'INALCO associée au CNRS, paris Septembre 1999.

[6] D. Grobgeld, "A Free Implementation of the Unicode Bidi Algorithm. 2003"

[7] E. Hart, "The Unicode Character-Glyph Model: What you Need to Know about Processing and Rendering Multilingual Text", 15th International Unicode Conference, San Jose, California, August30-September 2, 2004.

[8] A. Jacques, Caractères, codage et normalisation – de Chappe à Unicode, 2004.

[9] M. Leisher, "The UCData Unicode Character Properties and Bidi Algorithm Package." July 17, 2004.

[10] A. Patrick,. Introduction à Unicode et à l'ISO 10646. 2002.