Jurnal Teknologi, 49(D) Dis. 2008: 173–187
© Universiti Teknologi Malaysia

SHORT MESSAGING SYSTEM (SMS) COMPRESSION ON MOBILE PHONE-SMS ZIPPER

ALIAS MOHD[1] & WONG THAI MIN[2]

Abstract: In this paper we present a development of a new algorithm to develop a mobile application software that can compress and decompress multiple pages short message service (SMS) into fewer pages. This compressor/decompressor software (SMS Zipper) is programmed with Java 2nd Micro Edition (J2ME) language and implemented on Nokia Series 40 mobile phone. A novel compression/decompression algorithm has been developed to handle short text compression with twice the capacity compared to a standard message and can save the SMS cost of the users by compressing multiple pages SMS into fewer pages. Therefore, the compression of short messages to be conveyed over the wireless medium is very promising in terms of costs, time, capacity and bandwidth savings. It also may bring a new beneficial social impact to the development of linguistic by encouraging proper English usage in SMS messaging.

Keywords: SMS; SMS Zipper; text compression; J2ME

Abstrak: Di dalam kertas ini kami menerangkan pembinaan satu algoritma untuk membangunkan satu perisian aplikasi mudah alih yang boleh memampat/nyahmampat khidmat pesanan ringkas (SMS Zipper) yang berbilang-bilang muka surat kepada bilangan muka surat yang lebih padat. Perisian SMS Zipper ini diprogramkan dengan menggunakan Java Mikro Edisi Kedua (J2ME) dan boleh dipakai dalam telefon bimbit Nokia Series 40. Selain itu, satu algoritma pemampatan/nyahmampatan yang baru telah dibangunkan untuk mencapai objektif pemampatan teks yang pendek. Algoritma ini boleh memperuntukkan lebih kurang dua kali ganda muatan dibandingkan dengan mesej yang biasa. Dengan kejayaan aplikasi ini, kos SMS yang dibiayai oleh pengguna akan dapat dijimatkan dengan memampatkan SMS kepada bilangan muka surat yang lebih sedikit. Oleh itu, pemampatan mesej pendek yang dihantar oleh saluran tanpa wayar ini, amat berpotensi dari segi penjimatan kos, masa, pemuatan dan lebar jalur saluran. Selain itu, impak sosial yang lebih bermanfaat dalam pembangunan linguistik adalah dijangkakan dengan menggalakkan penggunaan bahasa Inggeris yang betul dalam SMS.

Kata kunci: SMS; SMS Zipper; pemampatan teks; J2ME

1.0 INTRODUCTION

Although SMS messaging has brought tremendous conveniences to everyone nowadays, there are still many problems hidden beneath this service. One of the main problems is the maximum number of characters for a single normal SMS is

[1&2] Information Technology Unit, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor Bahru, Malaysia
Email: aliasmohd@fke.utm.my

only 160 characters. Due to this limitation, many other problems have been arisen, such as:

- it would not be cost-effective if around 170 characters are sent. That is because the message will be sent as 2 SMSs (160 char. + 10 char.).
- if slightly more than 160 characters were typed in a message, most of the users would try to pack the message into one SMS by looking back the message and delete the "not important" words or change the words into abbreviation as many as possible. It would be a troublesome and time-consuming work.
- in order to save money by packing more information as much as possible into one SMS, many types of abbreviations has been used. The uncontrolled use of abbreviations in messaging has created a lot of strange messages which are hardly understood by the receiver without earlier synchronizing with the sender. Besides, this trend have arisen great impact to the development of socio-linguistic [3]. An example of a strange SMS which consists of various wierd abbreviations is shown in Figure 1.

Act, 2 f us x go 2 skol 2day 2. i m vr sry. N bec sg d. I ll cum agn tml. cul! byby!

$\downarrow$

Actually, 2 of us didn't go to school today too. I'm very sorry. Najeb back Singapore already. I will come again tomorrow. See you later! Bye bye!

Figure 1 Examples of abbreviated SMS and the original message

As an alternative to the problems of abbreviated SMS, a suitable compression/decompression algorithm for SMS messaging is extremely desired. This paper will describe the development of an algorithm to develop a SMS compression/decompression for mobile application with the following key features:

1.1 Suitable Algorithm to Compress and Decompress SMS

The algorithm must be simple, small space-consuming and fast in execution in a mobile device. The average compression efficiency of the message must be more than 40 percent. The efficiency is obtained by calculating the ratio of total characters in a compressed SMS to the total characters of the original SMS.

1.2 A Mobile Application Software to Compress and Decompress SMS Messages using the Developed Algorithm

The transmission of compressed data over cellular networks is done in a transparent way (so still not more than 160 characters or 140 bytes are transmitted in one SMS) and therefore the compression/decompression mobile application is needed on both ends as shown in Figure 2. That is because the compressed SMS will be encoded into fuzzy characters and cannot be understood by the receiving mobile device which does not have the same decompression application.

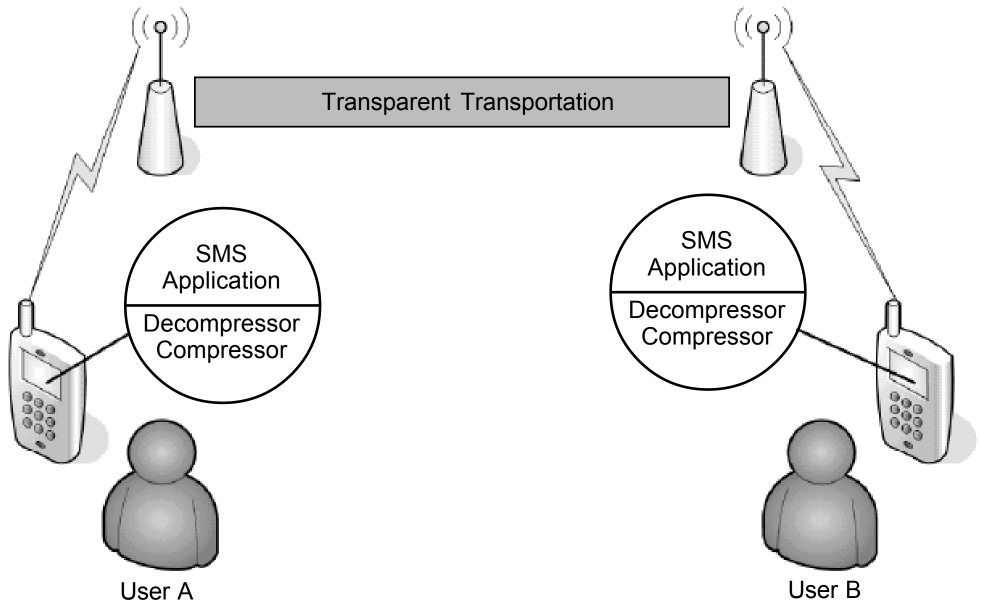


Figure 2 SMS compression and decompression application

1.3 A Cost Effective Application for the SMS Users

With this application the users can reduce significantly the number of single messages required for sending a long message. When the user have a long message to send, the application can be used to compress it and by sending the compressed message, shorter than the original, it will reduce the cost. In fact the user can send more information in a single page message.

For the application development, the mobile application using J2ME language is developed [5 – 9]. Nokia S-40 Series mobile phone is chosen as the platform of the software development. A Nokia S-40 Series Software Development Kit (SDK) [13], called Carbide.j 1.5 [12] is used to develop the mobile application. It is a user-friendly software with complete set of user's guides and tutorials. The mobile application is developed visually using this SDK by designing the graphic user interface (GUI), screen flow and programming with J2ME language. The application is designed to be used by J2ME Mobile Information Device Profile (MIDP) 2.0 enabled phones. Most of the mobile phones nowadays can support MIDP 2.0 and every mobile phone in Nokia S-40 Series is MIDP 2.0 enabled.

2.0 SMS COMPRESSION TECHNIQUES AND MESSAGE FORMAT

Many SMS compression methods have been developed to solve the limitation of 160 characters per message. These methods include human interpretive compression or so-called "word or phrase abbreviation," in which the compressed messages are manually keyed-in by the senders. Most of the compression software developed by the commercial corporate is based on these methods by programming the software to abbreviate the phrases or words automatically.

2.1 Human Interpretive Compression

In order to save the cost, many SMS senders tend to type as much information as possible within the limit of 160 characters per SMS by using various abbreviations. The abbreviations utilized in the message are popularly used and interpreted by the senders and receivers. Actually, this method can be classified as a kind of text compression which was known as "human interpretive compression." According to [1], the SMS coding methods can be summarized as below:

- **Truncation** (e.g: bro (brother), dy (already))
- **Phonetic respelling** (e.g: wif (with), cum (come))
- **Alphanumeric Homophony** (e.g: 2 (to), b4 (before), 29 (tonight))
- **Initialisation** (e.g: asap (as soon as possible))
- **Vowel Deletion** (e.g: pls (please), frm (from))
- **Logographic ASCII Emoticons** (e.g: :-) (happy), :-((sad))
- **Omission of spaces (optimization method)** (e.g: HiMarry, HowAreYouToday?)

2.2 Optimization Method

An SMS compression mobile application using optimization method removes spaces, deletes short words and concatenates long words. This method can only achieve averagely 20% of compression efficiency. An example of SMS message that was compressed using optimization method is shown in Figure 3 [11].

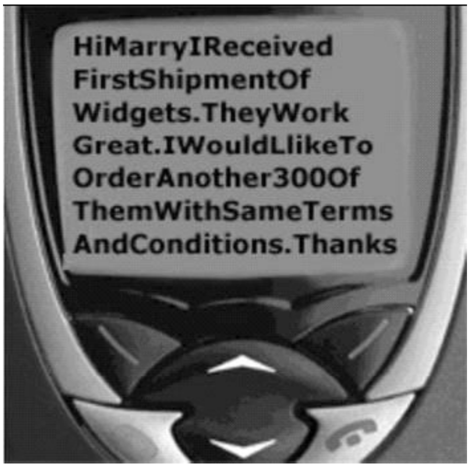

HiMarryIReceived
FirstShipmentOf
Widgets.TheyWork
Great.IWouldLlikeTo
OrderAnother300Of
ThemWithSameTerms
AndConditions.Thanks

Figure 3 Compression example of the optimization method

2.3 Abbreviation Method

According to [2], PC-based SMS software called MobiSMS for Outlook was developed by MobiMarketing (Mobile Marketing Solution). The compression methods used in this software include optimization method, carriage returns and punctuations removal, and convert words into abbreviations. SMS compression using this software allows the users to access the SMS dictionary to abbreviate their SMS message.

The compression methods that use words conversion and redundant characters omission to compress the messages are not efficient enough. The software that uses the method just replaces the manual compression by senders to automatic compression by mobile device or PC. The advantage of these applications is that the receiver mobile device doesn't need to have the decompressor installed, but the disadvantage is that the average compression efficiency is very low. Thus, a better and totally new concept of compression algorithm with better average efficiency is needed.

2.4 SMS Message Format

The SMS message, as specified by the European Telecommunications Standards Institute (ETSI) organization documents Global System for Mobile communications (GSM) 03.40 and GSM 03.38, can be up to 160 characters long [4], where each character is 7 bits according to the *7-bit default alphabet*. Eight-bit messages (max 140 characters) are usually not viewable by the phones as text messages; instead they are used for data in e.g. smart messaging (images and ringing tones) and Over-The-Air (OTA) provisioning of Wireless Application Protocol (WAP) settings. 16-bit messages (max 70 characters) are used for Unicode (UCS2) text messages, viewable by most phones. A 16-bit text message of class 0 will on some phones appear as a

Flash SMS (also known as blinking SMS or alert SMS). The method of sending SMS message by text mode and Protocol Data Unit (PDU) mode is described in [10].

3.0 COMPRESSION ALGORITHM

The development of the compression algorithm described in this paper consists of two methods, which are optimization and dictionary method. The combination of these two methods can achieve more than 40% of average compression efficiency. The combination of these two methods in one compression algorithm is found novel and has never been used in any SMS compression. Figure 4 shows the flow of SMS compression methods used in this paper.

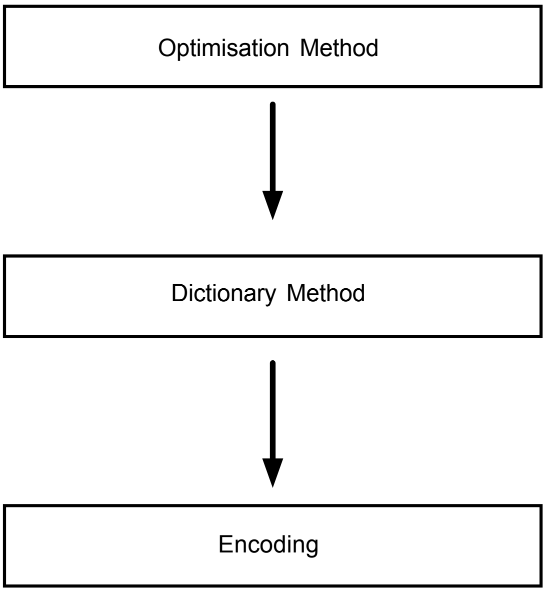


Figure 4 The flow of SMS compression

3.1 Optimization Method

In optimization method, all spaces will be omitted from the message. The first character of every word in the message will be capitalized and other characters will be converted into small capitals. This is to enable the decompressor to differentiate every word according to capitalized characters. In this method, the punctuation and carriage return omission are not included in the algorithm even though it can help in achieving better compression efficiency. That is because the decompressor in the receiver is not able to recover the positions of the punctuations and carriage returns omitted by the compressor in sender. Figure 5 shows the flow chart of optimization method.

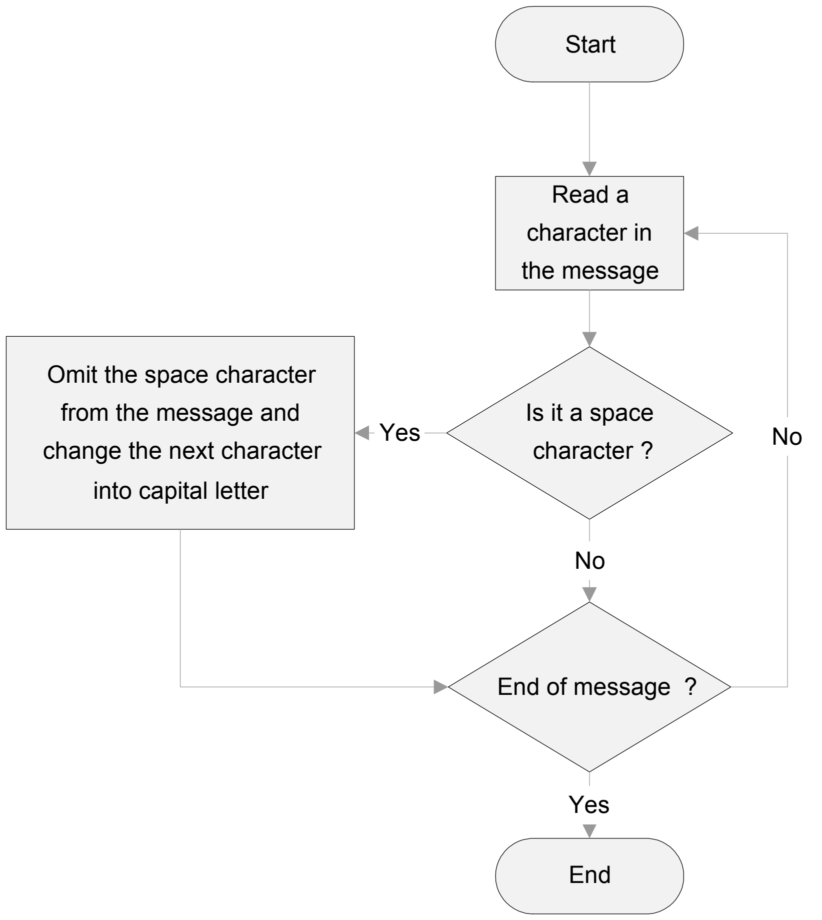

Start
Read a character in the message
Is it a space character ?
Yes
Omit the space character from the message and change the next character into capital letter
No
No
End of message ?
Yes
End

Figure 5 Flow chart of the optimization method

3.2 Dictionary Method

After the optimization method has completed, the dictionary method would be the second level compression. It is called dictionary method because a list of most frequently used English words and combination of characters are recorded in a module called dictionary. When the compressor found any of the word or combination of characters matches the one in the dictionary, it will be converted into an 8-bit special character from the range of Hex value 0×80 to $0 \times \text{FF}$. According to the GSM 03.38 specification, the content of the SMS message (user data) will be transformed into 7-bit (septet) data coding. If the dictionary method is used, 8-bit (octet) data coding will be used. Thus, the 160 characters per SMS limitation will be reduced to 140 characters because the maximum capacity in the user data is actually 140 bytes. However, the compression efficiency of using 8-bit data coding-cum-

Table 1 The ASCII characters table for dictionary method

Dec	Char	Dec	Char	Dec	Char	Dec	Char	Dec	Word	Dec	Word	Dec	Word	Dec	Word
0	@	32	<space>	64	¡	96	§	128	Back	160	ble	192	Am	224	ly
1	£	33	!	65	A	97	a	129	Also	161	But	193	am	225	Me
2	$	34	"	66	B	98	b	130	ance	162	Can	194	An	226	me
3	¥	35	#	67	C	99	c	131	Been	163	Com	195	an	227	ne
4	è	36	¤	68	D	100	d	132	From	164	Con	196	As	228	No
5	é	37	%	69	E	101	e	133	Give	165	Coz	197	as	229	no
6	ù	38	&	70	F	102	f	134	Have	166	day	198	At	230	Of
7	ì	39	'	71	G	103	g	135	Just	167	Did	199	at	231	of
8	ò	40	(	72	H	104	h	136	Last	168	Dun	200	Be	232	Ok
9	Ç	41	)	73	I	105	i	137	Like	169	Eat	201	be	233	On
10	<line feed>	42	*	74	J	106	j	138	Love	170	For	202	By	234	on
11	Ø	43	+	75	K	107	k	139	Many	171	Get	203	by	235	Or
12	ø	44	,	76	L	108	l	140	Meet	172	Got	204	De	236	or
13	<Cr>	45	-	77	M	109	m	141	ment	173	Had	205	Di	237	Pr
14	Å	46	.	78	N	110	n	142	Miss	174	Has	206	ea	238	ry
15	å	47	/	79	O	111	o	143	More	175	Her	207	ed	239	's
16	Δ	48	0	80	P	112	p	144	Need	176	His	208	en	240	Se
17	_	49	1	81	Q	113	q	145	Only	177	How	209	er	241	se
18	Φ	50	2	82	R	114	r	146	sion	178	ing	210	Go	242	Sh
19	Γ	51	3	83	S	115	s	147	That	179	ise	211	He	243	sh
20	Λ	52	4	84	T	116	t	148	Then	180	ive	212	he	244	So
21	Ω	53	5	85	U	117	u	149	They	181	Not	213	Hi	245	so
22	Π	54	6	86	V	118	v	150	Thru	182	Now	214	hi	246	ss
23	Ψ	55	7	87	W	119	w	151	tion	183	Our	215	Ic	247	sy
24	Σ	56	8	88	X	120	x	152	Very	184	Out	216	In	248	th
25	Θ	57	9	89	Y	121	y	153	Want	185	Say	217	in	249	To
26	Ξ	58	:	90	Z	122	z	154	Were	186	She	218	Is	250	to
27	{	59	;	91	Ä	123	ä	155	What	187	Sub	219	is	251	Un
28	Æ	60	<	92	Ö	124	ö	156	When	188	The	220	It	252	Up
29	æ	61	=	93	Ñ	125	ñ	157	Will	189	Was	221	it	253	up
30	ß	62	>	94	Ü	126	ü	158	With	190	Who	222	le	254	Us
31	É	63	?	95	§	127	à	159	Your	191	..	223	ll	255	us

7-bit default alphabets

Dictionary

compression algorithm is much higher than 7-bit data coding and more feasible to be implemented.

With this method the more experienced the user is, the better is the compression efficiency. That means if the user choose to type more words or combination of characters that have been predefined in the dictionary, more words will be encoded and the compression efficiency will be increased. So, the application that is developed will calculate the compression efficiency and provide the option for the users to choose whether they want to compress the selected message or not. If the compression efficiency is low, the user can choose to send the message without compression and vice versa.

Since the default 7-bit data coding is extended to 8-bit data coding, there will be 128 extra spaces (0 × 80 to 0 × FF) left. Thus 128 of words and combinations of characters that are most frequently used in English SMS messaging were chosen and recorded into the dictionary while the 7-bit default alphabets (0 × 00 to 0 × 7F) are remained. The dictionary table with a list of chosen English words and combinations of characters is shown in Table 1, while the flow chart of the dictionary method is shown in Figure 6.

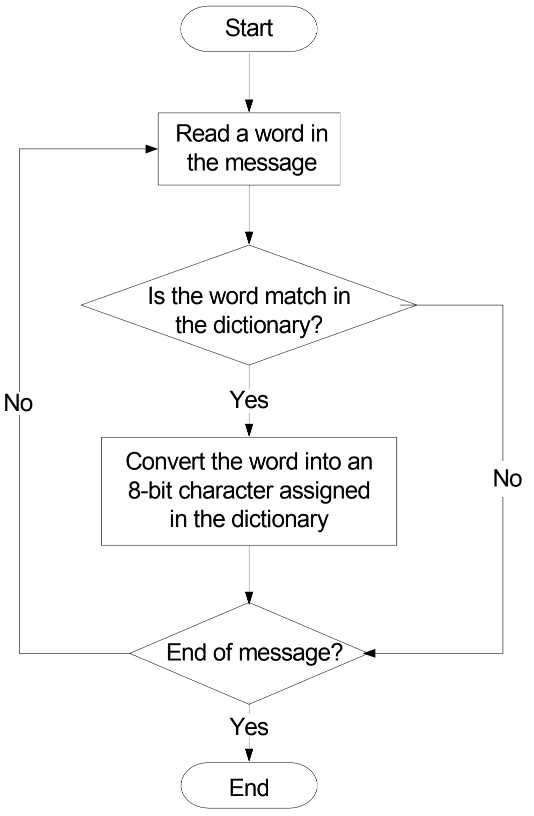
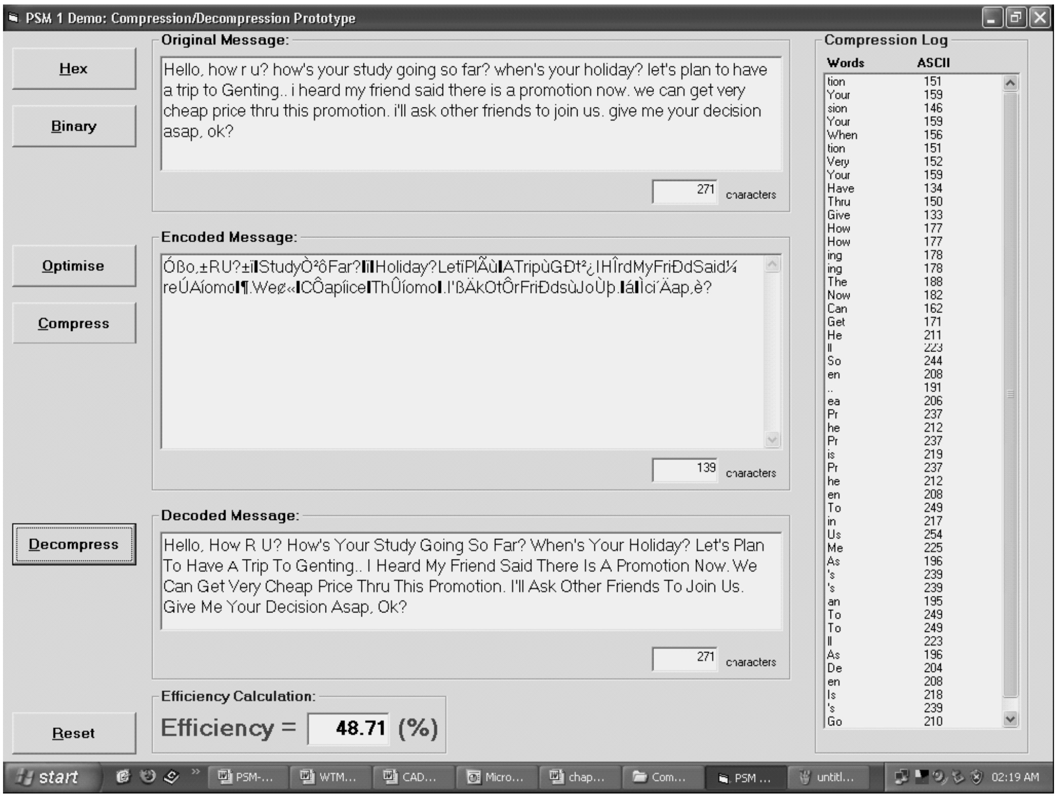

Start
Read a word in the message
Is the word match in the dictionary?
Yes
No
Convert the word into an 8-bit character assigned in the dictionary
No
End of message?
Yes
End

Figure 6 Flow chart of the dictionary method

4.0 RESULTS AND DISCUSSION

4.1 Prototype in Visual Basic

A Graphical User Interface (GUI) application software is programmed in Microsoft Visual Basic 6 to test and prove the functionality of the proposed compression/ decompression algorithm as shown in Figure 7.



Original Message:

Hello, how r u? how's your study going so far? when's your holiday? let's plan to have a trip to Genting.. i heard my friend said there is a promotion now. we can get very cheap price thru this promotion. i'll ask other friends to join us. give me your decision asap, ok?

271 characters

Encoded Message:

Óßo,±RU?±ï▌StudyÒ²ôFar?▌ï▌Holiday?LetïPlÃù▌ATripùGÐt²¿IHÎrdMyFriÐdSaid¼reÚAíomo▌¶.We¢«▌CÔapíice▌ThÛíomo▌.I'ßÄkOtÔrFriÐdsùJoÙþ.▌á▌Ìci´Äap,è?

139 characters

Decoded Message:

Hello, How R U? How's Your Study Going So Far? When's Your Holiday? Let's Plan To Have A Trip To Genting.. I Heard My Friend Said There Is A Promotion Now. We Can Get Very Cheap Price Thru This Promotion. I'll Ask Other Friends To Join Us. Give Me Your Decision Asap, Ok?

271 characters

Efficiency Calculation:

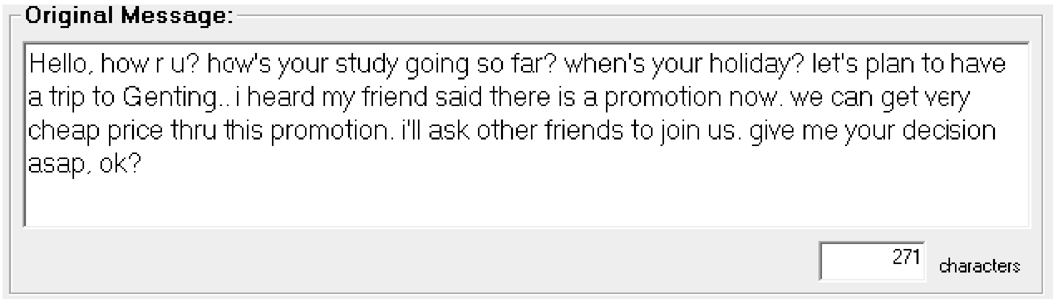
Efficiency = 48.71 (%)

Compression Log

Words	ASCII
tion	151
Your	159
sion	146
Your	159
When	156
tion	151
Very	152
Your	159
Have	134
Thru	150
Give	133
How	177
How	177
ing	178
ing	178
The	188
Now	182
Can	162
Get	171
He	211
ll	223
So	244
en	208
..	191
ea	206
Pr	237
he	212
Pr	237
is	219
Pr	237
he	212
en	208
To	249
in	217
Us	254
Me	225
As	196
's	239
's	239
an	195
To	249
To	249
ll	223
As	196
De	204
en	208
Is	218
's	239
Go	210

Figure 7 Compression/decompression prototype

The message is an example of SMS message which has 271 characters. If this SMS is stored in the mobile phone, it will be separated into 2 pages (160 char + 111 char). Figure 8 shows the example of an original SMS message in the prototype. The first level compression is done by using "Text Optimizing" method. All of the spaces in the messages are omitted and the first character of every word is converted



Original Message:

Hello, how r u? how's your study going so far? when's your holiday? let's plan to have a trip to Genting.. i heard my friend said there is a promotion now. we can get very cheap price thru this promotion. i'll ask other friends to join us. give me your decision asap, ok?

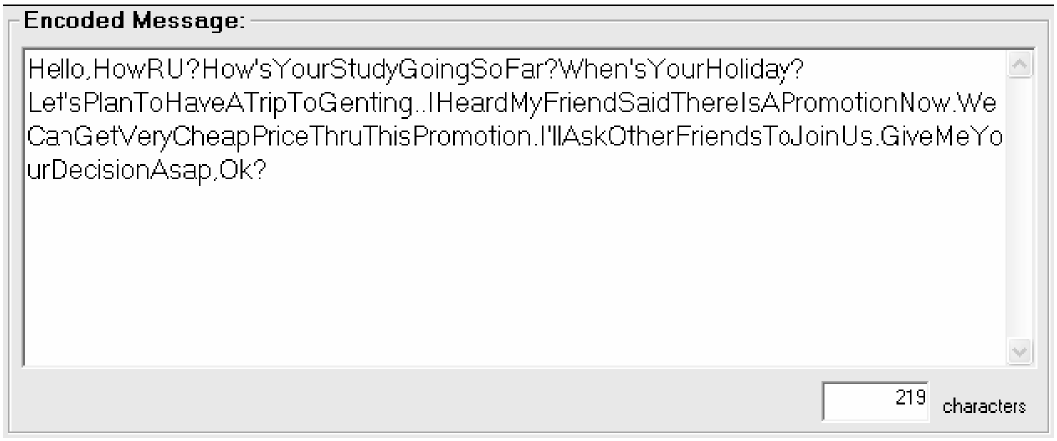
271 characters

Figure 8 Original message in the prototype

into capital letter. The total character in this first level encoded message has been reduced into 219 characters only. Figure 9 shows the optimized SMS message in the prototype.

The second level compression is done by using the "dictionary" method. Every combination of characters which matched the keywords in the "dictionary" will be converted to an 8-bit ASCII character with the value is from 128 to 255. The total character of this second level encoded message has been reduced to 139 characters only.

Figure 10 shows the compressed SMS message in the prototype. The compression efficiency of this message was calculated and the result was 48.71% as shown in Figure 11.

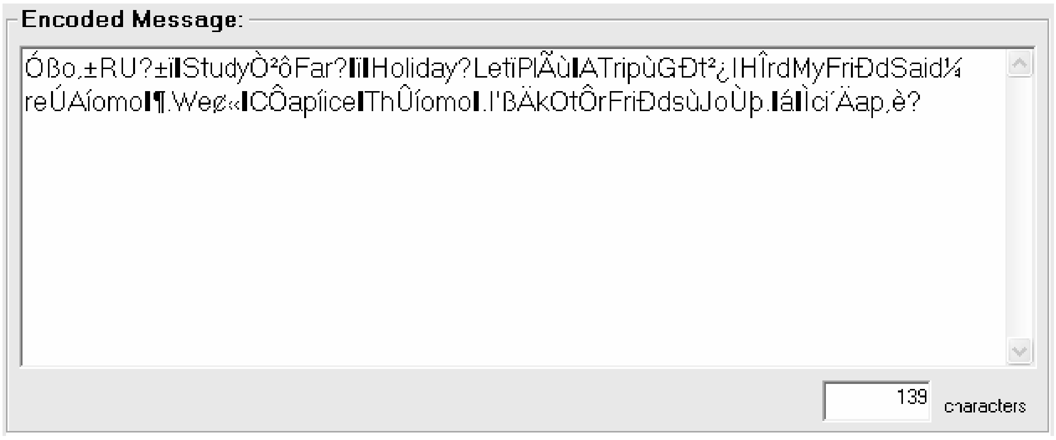


Encoded Message:

```
Hello,HowRU?How'sYourStudyGoingSoFar?When'sYourHoliday?
Let'sPlanToHaveATripToGenting..IHeardMyFriendSaidThereIsAPromotionNow.We
CanGetVeryCheapPriceThruThisPromotion.I'llAskOtherFriendsToJoinUs.GiveMeYo
urDecisionAsap,Ok?
```

219 characters

Figure 9 Optimised message in the prototype

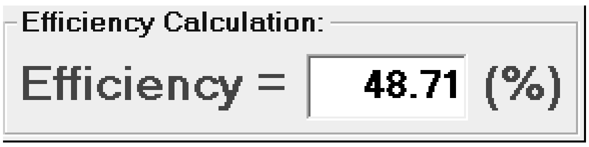


Encoded Message:

```
Óßo,±RU?±ï▌StudyÒ²ôFar?▌ï▌Holiday?LetïPlÃù▌ATripùGĐt²¿IHÎrdMyFriĐdSaid¼
reÚAíomo▌¶.Weø«▌CÔapíice▌ThÛíomo▌.I'ßÄkOtÔrFriĐdsùJoÙþ.▌á▌Ìci´Äap,è?
```

139 characters

Figure 10 Compressed message in the prototype


Efficiency Calculation:

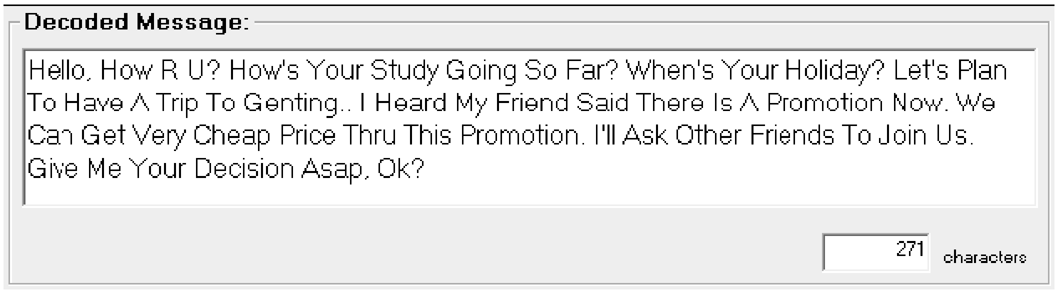
Efficiency = 48.71 (%)

Figure 11 Compression efficiency in the prototype

The formula of efficiency calculation is as below:

$$\text{Efficiency} = \frac{(\text{Total char in original SMS} - \text{Total char in compressed SMS})}{\text{Total char in original SMS}}$$

Then, the compressed message could be decompressed to achieve the original message as shown in Figure 12.


Decoded Message:

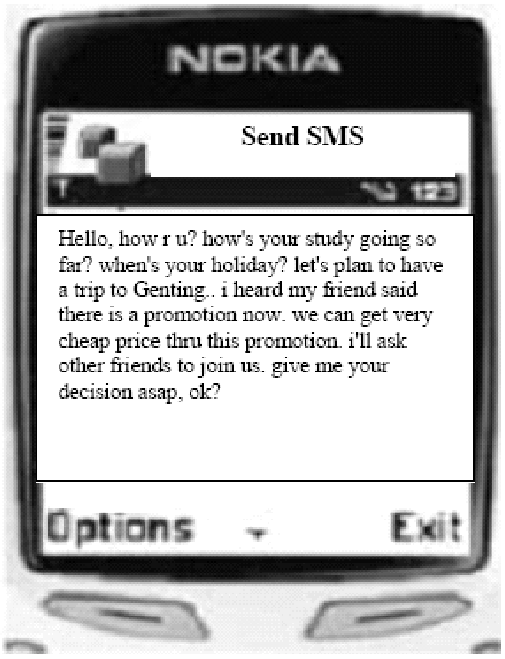
Hello, How R U? How's Your Study Going So Far? When's Your Holiday? Let's Plan To Have A Trip To Genting.. I Heard My Friend Said There Is A Promotion Now. We Can Get Very Cheap Price Thru This Promotion. I'll Ask Other Friends To Join Us. Give Me Your Decision Asap, Ok?

271 characters

Figure 12 Decompressed message in the prototype

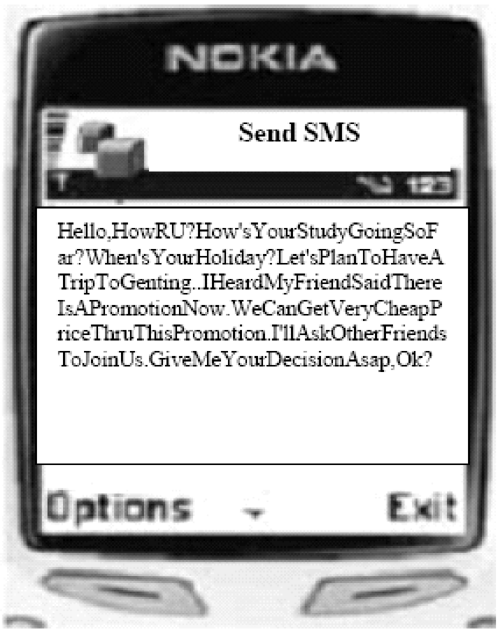
4.2 SMS Zipper in J2ME (ZMS)

After the completion of the prototype in Visual Basic, the applicable version of Java-based SMS compressor/decompressor software in J2ME is developed. A series of figures in the following pages shows the flow of the compression and decompression of the SMS message by the application in a mobile phone. Figure 13 shows an example of an original SMS message that was used to test the functionality of application. Figure 14 shows the output of the message after it was compressed using optimization method. Figure 15 shows the output of the message after it was compressed using dictionary method. Figure 16 shows the output of the message after receiver has decompressed the message received from the sender.



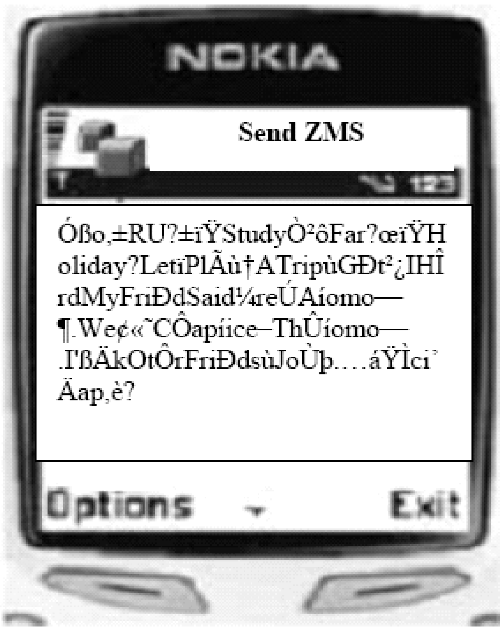
Hello, how r u? how's your study going so far? when's your holiday? let's plan to have a trip to Genting.. i heard my friend said there is a promotion now. we can get very cheap price thru this promotion. i'll ask other friends to join us. give me your decision asap, ok?

Figure 13 Original message of SMS



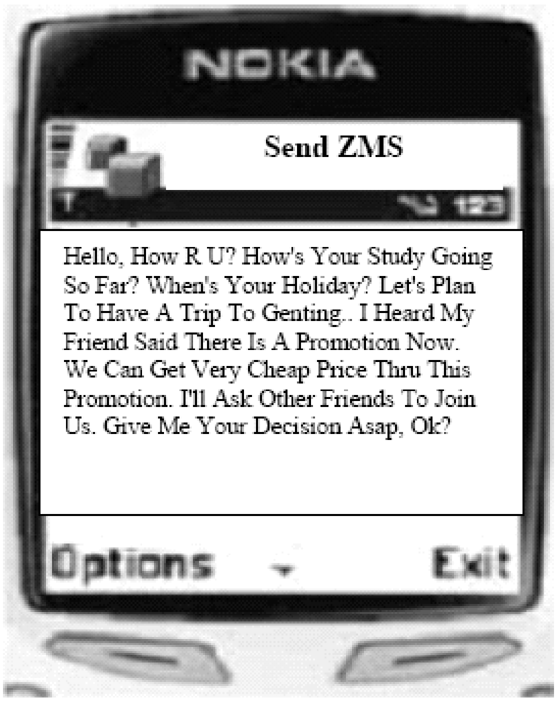
Hello,HowRU?How'sYourStudyGoingSoFar?When'sYourHoliday?Let'sPlanToHaveATripToGenting..IHeardMyFriendSaidThereIsAPromotionNow.WeCanGetVeryCheapPriceThruThisPromotion.I'llAskOtherFriendsToJoinUs.GiveMeYourDecisionAsap,Ok?

Figure 14 Output of message after optimization



Óßo,±RU?±ïŸStudyÒ²ôFar?œïŸH
oliday?LetïPlÃù†ATripùGĐt²¿IHÎ
rdMyFriĐdSaid¼reÚAíomo—
¶.We¢«˜CÔapíice–ThÛíomo—
.I'ßÄkOtÔrFriĐdsùJoÙþ….áŸÌci˚
Äap,è?

Figure 15 Output of message after compressed using dictionary method



Hello, How R U? How's Your Study Going So Far? When's Your Holiday? Let's Plan To Have A Trip To Genting.. I Heard My Friend Said There Is A Promotion Now. We Can Get Very Cheap Price Thru This Promotion. I'll Ask Other Friends To Join Us. Give Me Your Decision Asap, Ok?

Figure 16 Output of message after decompressed by the receiver

5.0 CONCLUSION

The SMS compression/decompression (SMS Zipper) mobile application system is successfully developed using the algorithm that is described in the paper. The system functions as a user friendly, cost effective and time saving mobile application to the users with the maximum efficiency of 48.71 percent based on several messages tested. The compression/decompression method that is invented can only support English language. That means only SMS with English contents can be compressed efficiently. So, a SMS compression method that can support other languages, such as Malay, Chinese and Tamil might be developed in the future. So far, the SMS Zipper that is developed in this program can only compatible with Nokia S40 Series mobile phones.

REFERENCES

[1] Dafydd, G., M. Kul. 2006. Economy Strategies in English and Polish Text Messages As Example of Channel Constraints. *Poznan Linguistic Meeting*. 20-22.

[2] MobiMarketing (Mobile Marketing Solution). Using SMS compression. http:*//www.mobimarketing.com/outlook/compress.htm* (accessed in March 2007).

[3] Rich, L. 2005. The Socio-Linguistic of SMS: An Analysis of SMS Use by A Random Sample of Norwegians. *Telednor Research Institut*. Norway: Oslo.

[4] Bodic, G. L. 2005. *Mobile Messaging Technologies and Services: SMS, EMS and MMS*. 2nd Edition. John Wiley & Sons.

[5] Scott, B., G. and M. J. Cronin. 2002. *Mobile Application Development with SMS and SIM Toolkit*. McGraw-Hill.

[6] Roger, R., A. Taivalsaari and and M. VandenBrink. 2001. *Programming Wireless Devices with Java 2 Platform Micro Edition*. Addison-Wesley.

[7] Tremblett, P. 2002. *Instant Wireless Java with J2ME*. Berkeley, California: McGraw-Hill/Osborne.

[8] Keogh, J. E.. 2003. *J2ME : The Complete Reference*. McGraw-Hill/Osborne.

[9] Kroll, M., 2001. *Java 2 Micro Edition (J2ME) Application Development*. Sams.

[10] SMS and the PDU format. *http://www.dreamfabric.com/sms/ (accessed in March 2007)*

[11] Send SMS Text Messages, SMS Compression. http:*//www.ipipi.com/help/compression.htm (accessed in March 2007)*

[12] Carbide.j 1.5 User's Guide. 2006. Nokia Corporation.

[13] Nokia Prototype SDK 4.0 for Java ME User's Guide. 2006. Nokia Corporation.