Tamil in Unicode

V Krishnamoorthy
(Former Professor of Anna University)
Inforeed (Information Research And Education)
30(4 C) Second Main Road, Gandhi Nagar, Adayar, Chennai 600 020 India

ABSTRACT
Some of the shortcomings of the present scheme for Tamil in Unicode is pointed out first. Two alternative schemes are presented here for Tamil for possible adoption in the Unicode. One uses 384 locations. The other uses the same 128 locations provided for Tamil, and extends the existing scheme by including the pure consonants. The advantages of these schemes are as follows. In the first scheme, the space requirement is just around 70% of what is needed in the current Unicode scheme. The manipulations needed for the NLP applications are enormously simplified and speeded up, as the vowel and consonant parts are available as blocks of bits in the code itself. The second scheme facilitates the splitting of a letter as consonant and a vowel in the right way. It will reduce the memory required by about 10%. No new space is required, and it is backward compatible with the current code. But this will not be as efficient as the first scheme.

1. Some shortcoming of the Tamil coding in the Unicode

2. The Tamil scholars have very aptly defined consonants and vowels. Except for the aaytham, the other Tamil letters are a combination of a consonant and a vowel. Unfortunately, the consonants do not find a place in this scheme. Instead consonants with the vowel a is included. This leads to the unnatural way of representing a pure consonant as a combination of one letter with another symbol pulli. It has been pointed out by others that this leads to problems while doing processing, due to its incorrect representation. Should we forget what was discovered thousands of years ago?

3. There is a character called ow length marker. Note that to represent the vowel ow, there is already a matra present in the scheme. What is introduced here as ow length marker is nothing but a glyph. But Unicode is supposed to code only characters and not glyphs! It should be noted that the reason behind this inclusion is differentiating this glyph from the glyph for the letter la. But one cannot differentiate between these two, as in today's writing, both the glyphs are identical. Even if we change them into two differently looking glyphs, a glyph should not find a place in the Unicode.

4. The explanations given for the creation of the glyphs for koo etc. are not worded properly. It gives the meaning as if the character koo is equivalent to kee and the thunai ezhthu.

5. The rendering of old type of letters for lai, raa etc. are provided. One has to recall that many years ago, the government of Tamil Nadu, by Government notification has changed all these. This old type rendering should not find a place in a current document.

6. There is nothing called anushwar in Tamil. This is used only in some other Indian languages, and not in Tamil. The Tamil code starts with this wrong symbol.

7. The aaythm is shown as a vowel modifier, by including a dotted circle, which is not correct. It is an independent letter.

2. Scheme 1

To make the use of Tamil very effective in terms of memory, it has been already proposed by a few that all the Tamil letters should be given individual positions in the Unicode scheme. The scheme proposed below is an extension of that thought. Whereas in such schemes of others, the letters are put one after other, in a continuos sequence, here we use a different approach. This leads to enormous simplicity in programming and processing.

The design of this scheme is influenced by the way the ASCII code is designed. In the ASCII code the lower case English alphabets are not put immediately following the upper case alphabets. It is put in such a way that the corresponding upper and lower case letters differ exactly in 32 positions. This will make the conversion of one from the other just by bit manipulation, which is faster than the table look up.

In the case of Tamil, this principle of 'use bit manipulations to speed up processing' can be used more effectively, as given below.

In Tamil, each consonant combines with 12 vowels. Including the pure consonant, there are 13 letters for each consonant. The aaytham and the 12 vowels also form 13 letters. There are 18 pure Tamil consonants and 5 grandha consonants. The grandha letter sri stands alone. So, there are 24 blocks of 13 letters each, and one single letter. Apart from these, the symbols for the numerals, and day, month etc. are to be accommodated. They number about 20. The arrangement of these letters and symbols, each block of 13 letters in one block of 16 places, is the basic idea in this scheme. Since there are 24 blocks exactly 128 x 3 = 384 positions are enough for the Tamil letters. In case we can get 128 x 4 = 512 places, the symbols can be accommodated after all the Tamil letters, and this will be the ideal scheme. In case we have to settle for 384 places, then, as in the case of ASCII, the remaining 3 positions in each block can be used for symbols,. The scheme is given below.

The code for a Tamil letter is given as follows. Assume that abcd efg0 0000 0000 is the starting location for the block provided for Tamil. Consider the position given by the 16 bit binary number abcd efgh ijkl mnop. Consider the Tamil letter, say kaa. This has the first consonant and the second vowel in it. Then efg0 0001 0010 gives the position of kaa. Here the binary number mnop (between 0 and 12) gives the vowel present in the letter. Also the binary number hijkl, which is between 1 and 23, gives the consonant present in that letter. If one of these numbers is zero then it represents a letter which is a pure vowel or a pure consonant. mnop = 13 and hijkl = 0 gives the position of the aaytham. mnop = 13 and hijkl = 23 gives the position of the the letter sree. The symbols for rupee, number, merpadi, date, month, year, debit, credit and Tamil numerals can be kept in the last row with mnop = 15.

In this scheme, the vowels come first. Then the aaythm comes. Then the uirmei letters of a particular mei follow that mei. this happens for the 18 + 5 consonants. then the letter sree comes. As in the case of the ASCII code for English, symbols come inbetween. Leaving these symbols, the Tamil letters come in the correct order. Note that since the uirmei letters depend on the mei letters, it is natural that the mei comes before the uirmeis.

The following is in the tabular form. The first 7 bits abcd efg are not shown here.

hijkl	0	1	2	3	4	5	6	7	8	9	10	11
mnop												
0		க்	ங்	ச்	ஞ்	ட்	ண்	த்	ந்	ப்	ம்	ய்
1	அ	க	ங	ச	ஞ	ட	ண	த	ந	ப	ம	ய
2	ஆ	கா	ஙா	சா	ஞா	டா	ணா	தா	நா	பா	மா	யா
3	இ	கி	ஙி	சி	ஞி	(r)	ணி	தி	நி	பி	மி	யி
4	ஈ	கீ	ஙீ	சீ	ஞீ	டீ	ணீ	தீ	நீ	பீ	மீ	யீ
5	உ	கு	ஙு	சு	ஞு	டு	ணு	து	நு	பு	மு	யு
6	ஊ	கூ	ஙூ	சூ	ஞூ	டூ	ணூ	தூ	நூ	பூ	மூ	யூ
7	எ	கெ	ஙெ	செ	ஞெ	டெ	ணெ	தெ	நெ	பெ	மெ	யெ
8	ஏ	கே	ஙே	சே	ஞே	டே	ணே	தே	நே	பே	மே	யே
9	ஐ	கை	ஙை	சை	ஞை	டை	ணை	தை	நை	பை	மை	யை
10	ஒ	கொ	ஙொ	சொ	ஞொ	டொ	ணொ	தொ	நொ	பொ	மொ	யொ
11	ஓ	கோ	ஙோ	சோ	ஞோ	டோ	ணோ	தோ	நோ	போ	மோ	யோ
12	ஔ	கௌ	ஙௌ	சௌ	ஞௌ	டௌ	ணௌ	தௌ	நௌ	பௌ	மௌ	யௌ
13	ஃ											
14												
15	Tamil symbols and Tamil numerals come in this row											

hijkl	12	13	14	15	16	17	18	19	20	21	22	23
mnop												
0	ர்	ல்	வ்	ழ்	ள்	ற்	ன்	ஸ்	ஷ்	ஜ்	ஹ்	க்ஷ்
1	ர	ல	வ	ழ	ள	ற	ன	ஸ	ஷ	ஜ	ஹ	க்ஷ
2	ரா	லா	வா	ழா	ளா	றா	னா	ஸா	ஷா	ஜா	ஹா	க்ஷா
3	ரி	லி	வி	ழி	ளி	றி	னி	ஸி	ஷி	ஜி	ஹி	க்ஷி
4	ரீ	லீ	வீ	ழீ	ளீ	றீ	னீ	ஸீ	ஷீ	ஜீ	ஹீ	க்ஷீ
5	ரு	லு	வு	ழு	ளு	று	னு	ஸு	ஷு	ஜு	ஹு	க்ஷு
6	ரூ	லூ	வூ	ழூ	ளூ	றூ	னூ	ஸூ	ஷூ	ஜூ	ஹூ	க்ஷூ
7	ரெ	லெ	வெ	ழெ	ளெ	றெ	னெ	ஸெ	ஷெ	ஜெ	ஹெ	க்ஷெ
8	ரே	லே	வே	ழே	ளே	றே	னே	ஸே	ஷே	ஜே	ஹே	க்ஷே
9	ரை	லை	வை	ழை	ளை	றை	னை	ஸை	ஷை	ஜை	ஹை	க்ஷை
10	ரொ	லொ	வொ	ழொ	ளொ	றொ	னொ	ஸொ	ஷொ	ஜொ	ஹெ	க்ஷொ
11	ரோ	லோ	வோ	ழோ	ளோ	றோ	னோ	ஸோ	ஷோ	ஜோ	ஹே	க்ஷோ
12	ரௌ	லௌ	வௌ	ழௌ	ளௌ	றௌ	னௌ	ஸௌ	ஷௌ	ஜௌ	ஹெ	க்ஷௌ
13												
14												

15 Tamil Symbols and Tamil Numerals come in this row

The advantages of this 384 place scheme are enumerated below.

1. In Tamil, a letter can be a pure vowel, a pure consonant, or it can be formed by combining a consonant and a vowel. Also, a word is formed by combining many parts, sometimes as much as eight parts. In natural language processing, while splitting a word to get its components, the splitting rules, which are the reverse of the combining rules, often needs information of the following type, about a letter.
1. Whether a letter is a pure vowel?
2. Whether a letter is a pure consonant?
3.Whether a letter is a ukaram?
4. Whether a letter has a particular consonant?
5. Whether a letter has a hard consonant?
6. Whether a letter is the soft consonant pair of a given hard consonant?

In all these cases, the answers can be got just by checking some bit positions, since both the vowel number and the consonant number are directly available as blocks of bits in the code itself. This will lead to enormous simplification in the programming and also in the Natural language processing. Note that NLP is going to be the hot topic in the coming years. And this may effectively seal the fate of many languages. Note that in the case of table look up it may take more time for all processing.

2. In NLP, many times it is necessary to combine a vowel and a consonant, or to split a letter into its component vowel and consonant. This can also be achieved by bit manipulations.

3. The memory needed to store any text will be just around 60% of what will be needed if the existing Unicode is used. Obviously the communication time required also becomes just half.

4. If 512 locations are used, no special algorithm is needed to sort in the Tamil order. In the case of 384 places, if the special Tamil symbols are not used in the text, any normal sorting algorithm will work without any modification. In case the special symbols are also used, then they may have to be separated first, and then the text and the symbols have to be sorted using any commonly available algorithm, and then they have to be combined. Since in most of the cases, the symbols are not used, any sorting algorithm will work. This will be a boon since any data base software can be used even for Tamil sorting, without any modification.

1. It is shown that the proposed scheme scores over all the other schemes, in terms of memory, communication speed, and processing speed. These three are very important factors which may be crucial in determining whether a language is going to be a language used effectively in the future.

3. An alternative scheme

We have already noted that the ow length marker at position 0BD7 is not necessary. It is not at all a character, and hence can be safely removed from the code. This gives 25 continuous vacant places from 0BCE to 0BE6. The pure consonants 18 + 5 + 23 in number can be accommodated in this slot, say, starting from position 0BCE. The Grandha letter sree is treated as a single character in Tamil. As such one position has to be given for that, after these 23 places.

This will avoid the problem in processing. Also the both the pure consonants and the half consonants (those with a) can be represented by a single codes. This may save about 10% of the space. Backward compatibility will be maintained. Though this is obviously not the best choice, this is also being proposed purely because of its backward compatibility.

4. Conclusion

Some of the drawbacks of the preset Unicode scheme for Tamil has been pointed out. One scheme proposed uses 383 places and is shown to be best suited when we consider the future of Tamil in a competitive world. Another scheme is presented which is an extension of the current scheme. Though this is not the best, it offers backward compatibility, and solves one important problem. The Tamil community should take a decision after carefully considering the future requirement of Tamil processing in a highly competitive global scenario.

Tamil Encoding in Unicode - A Comparative Study

P.Chellappan
Palaniappan Bros., 14, Peters Road, Chennai - 600014
<Email: chellappan@vsnl.com>

INTRODUCTION

During the TamiNet99 Conference, which was held in Chennai, several papers were presented regarding the need to change the present Encoding of the Tamil Script in Unicode that occupies the Unicode locations U+0B80 to U+0BFF. Among the people who wanted a change, there were two schools of thought. There was one group that wanted assignment of unique locations only for the Uyir, Ayutham, Mei and Grantha characters (12+1+18+5+1). The other group wanted allocation of space for all the 313 Tamil characters. The author of this paper had also presented a paper calling for an encoding scheme that has a unique allocation for each of the 313 Tamil characters (Uyir, Mei and Uyir-Mei including the Grantha characters) and also for other Tamil Symbols. The purpose of this paper is to make a comparative study of the three different encoding schemes, so that a decision can be taken at the earliest.

THREE SCHEMES

1) THE PRESENT UNICODE SCHEME

This scheme allocates unique locations for the Uyir, Akaram Eriya Mei, Vowel Modifier and Symbol. Instead of treating an Uyir-Mei character as a combination of a Mei character and an Uyir character, it is treated as a combination of an Akaram Eriya Mei character and a Vowel modifier character. It also treats the Tamil Grantha characters 'ksha' and 'sri' as conjunct consonants. In order to be compatible to the other Indic scripts, the allocation of these characters are not as per the Tamil sort order. 128 locations are sufficient for this scheme.

Advantages:
All the Indic languages are allocated a block of 128 locations each and similar characters occupy the same relative location within the block. This enables easy transliteration possible between the Indic languages. Just a relative shift of locations would be sufficient for transliteration from one Indic language to another.

It helps in Natural Language Processing, Spell Check etc since the Uyir-Meis are already split into its basic components.

Disadvantages:
Since the Uyir-Meis are represented as combinations of an Akaram Eriya Mei and a Vowel Modifier each one of these characters would take up 32 bits (16 bits each for the Akaram Eriya Mei and Vowel Modifier characters). This results in large file sizes and also poor efficiency in processing.

Because of the same reason, there is no 1:1 relationship between characters and glyphs. Hence Glyph substitution will be required for proper display rendering. Tamil cannot be implemented in Level 1 of Unicode like English and the CJK (Chinese, Japanese and Korean) languages.

It ignores the natural sort order of the Tamil script. Hence it requires a separate Weight Table for proper sorting.

Only softwares that are Tamil enabled can be used.

This scheme does not follow the proper Tamil Grammatical rules.

2) PROPOSAL 1

In this scheme unique locations are allotted only for the Uyir, Mei, Grantha and symbol characters. The proper grammatical structure of Tamil is implemented in this scheme. All Uyir-Mei characters are represented as combinations of a Mei and a Uyir character. 128 locations are sufficient.

Advantages:
It helps in Natural Language Processing, Spell Check etc since the Uyir-Meis are already split into its basic components.

It maintains the Grammatical Structure of the Tamil language.

Since the proper sort order is maintained while allocation itself, straightforward sorting, without the need for a separate sort table, is possible.

Disadvantages:
File sizes are large since the Uyir-Meis are treated as combination character of Mei and Uyir character. This in turn leads to poor efficiency.

Since there is no 1:1 relationship between Characters and Glyphs, Level 1 implementation of Unicode is not possible.

Only softwares that are Tamil enabled can be used.

Transliteration to other Indic languages is slightly more difficult than the existing scheme.

3) PROPOSAL 2

This proposal envisages allocation of unique locations for each of the 313 Tamil characters and all the required Tamil Symbols. In this scheme all Uyir, Mei and Uyir-Mei characters including the Grantha characters are represented as single 16 bit characters (Unicode Characters) and not as combinations of Mei and Uyir characters. This proposal will require increase of the number of locations assigned for the Tamil script from 128 to 313+.

Advantages:
Since all the characters are represented only as 16 bit characters, the file sizes are smaller and as a result it is more efficient.

There is a perfect 1:1 relationship between characters and glyphs. Hence Tamil can be implemented even in a software that is Unicode Level 1 compliant. Literally all available softwares can be used for Tamil, without difficulty.

Sorting is easy and there is no need for separate Sort Weight Tables.

Disadvantages:
Since all Uyir-Meis are stored as single characters, one will have to use a mathematical manipulation to split it into its Mei and Uyir component. Hence at a first glance one will be led to believe that it is not suited for Natural Language Processing, Spell Check etc., But since efficiency is lost only in a memory operation as opposed to the loss of efficiency in a storage device read/write operation, this scheme still results in a better performance than the first two schemes.

Transliteration to other Indic languages is slightly more difficult than the existing scheme.

TESTING

The above comparison of the three schemes clearly shows that the all characterrepresentation scheme (Proposal 3) is the best. However all the theoretical discussions will have to be verified by proper testing.

Since both the Existing Scheme and Proposal 1 encode the Uyir-Mei characters as combination characters, efficiency of both these schemes would be similar except maybe in Natural Language Processing, Spell checking etc., where Proposal 1 could be better.

Hence as a matter of convenience, testing was done only to compare Proposal 1 and Proposal 2.

METHODOLOGY

As a preliminary testing process, a Pseudo Testing scheme was designed. A sample text of 25 pages was taken from an existing book and it was re-encoded according to Proposal 1 and Proposal 2 as show below.

Encoding:
Proposal 1 : Each Uyir and Mei character was encoded as a series of two bytes (8x2). The first byte would contain the Uyir or Mei character and the second byte would be blank.

e.g. அ = அ_ and க் = க்_

Each Uyir-Mei character was encoded as a series of four bytes (8x4). The first pair of bytes (16 bits) contains the Mei character and the second pair (16 bits) contains the Uyir character.

e.g. கி = க்_இ_ and = ச்_ஒ_

Proposal 2 : Each Uyir and Mei character was encoded exactly as in Proposal 1.

e.g. அ = அ_ and க் = க்_

However each Uyir-Mei character was encoded only as a series of two byte (8x2) characters.

e.g. கி = க்இ and சொ = ச்ஒ

The above two pseudo encoding schemes simulate the real situation fairly well.

The text derived from the above re-encoding process was used for testing various parameters that would affect the efficiency of the two schemes. For this purpose the following tests were carried out:

1. File size
2. Compressed file size using Pkzip
3. File copy using windows copy command (100 times)
4. Database Sorting of words from the text (20 times)
5. Database Indexing of words from the text (20 times)
6. Full word search for 'அவர்' in the complete text
7. Search for characters 'அன்' in any word in the complete text. e.g. in அவன்

TEST RESULTS
The results obtained from the above tests are tabulated below :

Sl.	Test	Proposal 2	Proposal 1	Difference
1.	File Size	116394 bytes	173904 bytes	49.41 %
2.	Compressed	35917 bytes	39467 bytes	9.88 %
3.	File Copy	1540 msecs	2080 msecs	35.06 %
4.	Database Sort	2310 msecs	3020 msecs	30.74 %
5.	Database Indexing	5490 msecs	7910 msecs	44.08 %
6.	Full word search	38450 msecs	58220 msecs	51.42 %
7.	'அன்' search	38010 msecs	57900 msecs	52.32 %

The pseudo test results are very clearly in favour of Proposal 2.

Other Languages: The concept of encoding all characters even if they are syllables, has been utilised by many languages. Primary examples are the Japanese Hiragana and Katakana script and the Korean Hangul Syllable block.

The Hiragana and Katakana Script allocates separate locations for syllable characters. e.g. 'ka', 'ki', 'ku', 'ke', 'ko', 'sa', 'si', 'su', 'se', 'so', and 'ta', 'ti', 'tu', 'te', 'to'.

Similarly, the Hangul Syllable block allocates a different location for each one of its syllables that are either a consonant-vowel-consonant combination or a consonant-vowel combination. In fact there are over 11172 such syllables which are allotted individual locations in Unicode (U+AC00 - U+D7A3). Apart from this the Hangul script also has a separate block called Hangul Jamo Blocl (U+1100 - U+11FF) which encodes the consonants and vowels alone without its combinations.

Another point to be noted is that the Canadian Syllabics have been alloted over 700 locations in Unicode 3.0

CONCLUSION:

Preliminary pseudo test results point clearly towards the All Character Encoding Scheme. But before proceeding further, it is necessary to test it out in the actual Unicode environment. This would require development of fonts and keyboard drivers. For this purpose the Tamils could come to a private understanding and use the End User subarea of the Private Use Area of Unicode (U+E000 - U+F8FF) for encoding all the Tamil characters. Once this testing is done, we would be in a position to take a final decision about how to proceed further. In case the results favour an All Character Encoding scheme, we should press further and get this implemented through the Unicode Consortium.

Author : The author is a partner of M/s Palaniappa Bros., which is one of the leading Tamil book publishing houses in Tamil Nadu. He is a Production Engineer with a Masters degree in Business Administration specialising in Finance and Information Systems. He has been involved in the fields of Font and Software development and DTP for over 15 years.

Contact : Palaniappan Bros.
14, Peters Road, Chennai - 600014, India.
Phone : 91-44-8268035, Fax : 91-44-8284067, eMail : chellappan@vsnl.com

தமிழ் மின்னிதழ்கள்

சி. அண்ணாமலை
Chennai

சிங்கப்பூரில் அக்டோபர் 1995-ம் ஆண்டு தமிழை முதலில் இணையத்தில் ஏற்றிய நா.கோவிந்தசாமியும் டாக்டர் டான் டின் வீவும் லியோங்கோக் யாங்கும் எதிர்கால தமிழ் இணையம் பற்றி என்ன நினைத்தார்கள் என்று தெரியாது. ஆனால் மிகக்குறுகிய காலத்தில் தமிழ் இணையம் அசுரமாய் முன்னேறிவிட்டது.

உலக அளவில் இணையத்தில் உயர்ந்துநிற்கும் சிலமொழிகளில் தமிழும் ஒன்று. உலகெங்கும் புலம்பெயர்ந்தும் வேலை நிமித்தமும் சிதறிக்கிடக்கும் தமிழர்கள் மொழியை நேசிப்பதன் காரணமாக நிறைய தகவல்கள் இணையத்தில் கிடைக்கின்றன. அதனால் இணையத்தில் ஏறிய முதல் இந்திய மொழி தமிழ் என்பதுடன் அதிகத்தகவல்களை கொண்ட இந்திய மொழியும் தமிழ்தான் என்றானது.

இந்தநேரம் தமிழ்க்கணினிக்கும் இணையத்திற்கும் பங்காற்றிய அனைவரையும் பாராட்டவேண்டும். அவர்களை கௌரவிக்காவிட்டாலும் தமிழ் உலகத்திற்கு அவர்களை அறிமுகப்படுத்தலாம். வேறெந்த மொழியிலும் நடைபெறாத ஒரு மொழிச்சேவை நடந்துள்ளது என்பதுதான் இதை சொல்லத்தோன்றுகிறது.

மறைந்த எழுத்தாளர் நா.கோவிந்தசாமி தமிழை மிகவும் நேசித்த, பெரும் மனிதாபிமானி. அவர் கணினி, இணையத் தொழில்நுட்ப நிபுணர் என்பது தமிழின் அதிர்ஷடம். இவரது பங்களிப்பு தமிழ் கணினி - இணையத்திற்கும் குறிப்பிடத்தக்கதாகும். தமிழை முதலில் இணையத்தில் ஏற்றிய பூரிப்பால் பெருமைபேசிகொண்டிருக்காமல் தமிழுக்குச் செய்யவேண்டிய கடமைகளை உணர்ந்தார். தனது சொந்த செலவில் சிங்கையிலிருந்தபடியே சென்னையிலும் ஓர் அலுவலகம் திறந்து பல மின்னிதழ்கள் நடத்தினார். நிதிச்சுமை அழுத்தினாலும் இதழ்களின் தரம் பற்றியே பேசுவார். சிறுபிழை என்றாலும், செய்திகள் இடம்பெற தாமதமானாலும் சென்னைக்கு பேசுவார். அவரது ஆழ்ந்த ஈடுபாடும் உழைப்பும், அர்ப்பணிப்பும் அவரை தமிழ் இணையத்தின் முன்னோடியாக்கியுள்ளது. தமிழ்க்கணினி, இணையம் பற்றிப்பேசும்போது அவரை எப்படி மறக்க முடியும்? இப்படி பலர் பணிசெய்கிறார்கள்.

தமிழ் இணையத்தில் ஏறிய உடன் கணியன், தமிழ்சினிமா, இன்தாம்.... போன்ற வலையகங்கள் உருவாகின. இன்று ஆறாம்திணை, அம்பலம், தமிழ்-தமிழா, திண்ணை, வானவில், தமிழ், தமிழ்நெட், வெப்உலகம், மலேசியநண்பன்... என பல தமிழ் மின்னிதழ்கள் நடத்தப்படுகின்றன.

ஆரம்பத்தில் அச்சு ஊடகத்தன்மையோடு அப்படியே பக்கங்கள் இடம்பெற்றன. எழுத்து, வடிவமைப்பு வகையிலும் சிறப்பாக இல்லை. ஆனால் இணையத்தில் படிப்பது, இணைப்புகளைப் (LINKS) பயன்படுத்துவது, ஒரே நேரம் பலபேர் படிப்பது, பிழைகளை மீண்டும் திருத்தமுடிவது, பக்கங்கள் பிரச்னையாக இல்லாமல் இருப்பது, உடனடி கருத்துகளை அறிய முடிவது... போன்றவை பிரமிப்பாக இருந்தன. மேலாக, புது ஊடகம் என்பதால் பலரையும் ஈர்த்தது.