Designing packages for Λ: An overview

Apostolos Syropoulos

1 Introduction

The Ω typesetting engine was introduced about ten years ago [2]. Roughly speaking, it is a Unicode TEX extension, and, in our opinion, it is the best TEX extension available. But even after ten years, there are certain things which remain undocumented. For example, there is no single document describing how one can prepare a Λ package! However, this lack of documentation is somehow justified since Ω is still an evolving system. Thus, sometimes it makes no sense to document an experimental feature that may not be present in the next release of the system. Still, there are certain features that are frozen and so we need proper documentation for at least these features.

Generally speaking, any LATEX package is a Λ package, but the inverse does not hold. The previous assertion is true just because Ω provides the so-called Ω Translation Processes (ΩTP, for short), which are used to transform the character encoding of the input stream and are among the novelties introduced in Ω. Thus, new Λ packages, whose functionality relies on ΩTPs, should provide the "familiar" user-interface (i.e., thru package options and commands) for their activation/deactivation.

It is our belief that Λ packages should accept Unicode encoded files just like TEX accepts ASCII files. In practice, this means that we need Unicode-encoded fonts. Since Type 1 fonts do not provide such a facility, and the use of OpenType fonts is still an item of active research, we need to create Ω virtual fonts in order to create virtual Unicode fonts.

In this paper, we begin by presenting a few particulars about the Inuit people and their language. Next, we present the general functionality of the oinuit package. We continue with the implementation details of the package. In particular, we describe how we implemented the various package options, the language switching commands, and the design of the various ΩTPs involved. In addition, we outline the implementation of Ω virtual property list files, which are used to create virtual fonts, and we finish with a description of the implementation of the hyphenation rules of the Inuktitut language.

2 Inuits and their language

The Inuit (here we also include the closely-related Yupik) are a native people of the Canadian Arctic, Greenland, Alaska, and the Chukotka Autonomous Okrug of the Russian Federation. Inuktitut (the language of the Inuit) and Yupik together form the Eskimo branch of the Eskimo-Aleut language family. The Eskimo branch is estimated to have 73000 speakers at present. Although linguists continue to use the term Eskimo, the people themselves prefer the term *inuit*, which is the plural form of the word *inuk*, meaning "human being".

Morphosyntactically, Inuktitut is an agglutinative or polysynthetic language. This means that multiple morphemes combine into what can be called words, which represent concepts that may require entire sentences in other languages. Inuktitut also features a morphological process called incorporation. A fuller discussion of these topics is beyond the scope of this paper.

James Evans, a Wesleyan (Methodist) missionary, is the creator of the Inuit syllabary. This writing system was initially created for the Ojibwe language, based on Pitman shorthand. Later Evans learned the Cree language and adapted his syllabary to write a translation of the New Testament in the Cree language. Rev. Edmund Peck adapted Evans syllabary and introduced it to the Inuit people at Little Whale River in 1876.

It is interesting to note that the syllabics are used by Inuit who live in Canada, especially in the new Canadian territory of Nunavut. On the other hand, Inuit in (what is now) the Northwest Territories, Labrador Coast and in Alaska use the Roman alphabet, as do the Inuit of Greenland (Greenlandic). Siberian Inuit use the Cyrillic script to write Inuktitut. Unfortunately, the use of the Inuktitut language has declined in those areas where syllabics are not used (with the lone exception of Greenland). In Table 1 the reader can view the Inuktitut syllabary currently in use as well as the Latin transcription of each symbol. For more information on the history of the Inuktitut syllabary the reader should consult the excellent article by Kenn Harper [1].

3 Typesetting Inuktitut with Λ

We now briefly describe the functionality of the oinuit package.

The package provides five options: `nunavut` (default option), `quebec`, `iscii`, `utf8`, and `ucs2`, which respectively correspond to source text using the Latin transcription of Inuktitut and the Anglican orthography, the Latin transcription of Inuktitut and the Catholic orthography, the Inuit ASCII (see Table 2), the UTF-8 Unicode encoding, and the UCS-2 Unicode encoding. Note that the Inuit ASCII is based on a PC Inuit Character Table proposed by Everson Typography (see the page at `www.evertype.com/standards/iu/iu-tables.html`).

ᐃ	i	ᐅ	u	ᐊ	a	ᕼ	h
ᐱ	pi	ᐳ	pu	ᐸ	pa	ᐦ	p
ᑎ	ti	ᑐ	tu	ᑕ	ta	ᑦ	t
ᑭ	ki	ᑯ	ku	ᑲ	ka	ᑲ	k
ᒋ	gi	ᒍ	gu	ᒐ	ga	ᒡ	g
ᒥ	mi	ᒪ	mu	ᒪ	ma	ᒻ	m
ᓂ	ni	ᓄ	nu	ᓇ	na	ᓐ	n
ᓕ	li	ᓗ	lu	ᓚ	la	ᓪ	l
ᓯ	si	ᓰ	su	ᓴ	sa	ᔅ	s
ᔨ	ji	ᔩ	ju	ᔭ	ja	ᔾ	j
ᕆ	ri	ᕈ	ru	ᕋ	ra	ᕿ	r
ᕙ	vi	ᕚ	vu	ᕝ	va	ᖁ	v
ᖃ	qi	ᖅ	qu	ᖄ	qa	ᖅ	q
ᖏᒋ	ngi	ᖏᒍ	ngu	ᖏᒐ	nga	ᖏ	ng
ᖧ	lhi	ᖨ	lhu	ᖦ	lha	ᖩ	lh
ᖓᒋ	nngi	ᖓᒍ	nngu	ᖓᒐ	nnga	ᖓ	nng

Table 1: The Inuit syllabary and its Latin transcription.

Also, note that when using the Anglican orthography, one places a dot over a symbol to denote that the vowel of that syllable is "long"; whereas when using the Catholic orthography, the difference in vocalic length is indicated by duplicating the symbol for the vowel which is long.

To assist people who happen to use an ASCII editor to prepare their documents, we defined a few commands that can be used to switch languages and fonts. The command `\textinuit` assumes that its argument is a piece of Inuktitut text that is typeset accordingly. Similarly, the command `\inuittext` changes the internal state of Λ and everything from now on is assumed to be Inuktitut text. The environment `inuit` does exactly what the command `\textinuit` does. In addition, it is possible to switch between languages with the `\selectlanguage` command. Notice that all these commands implement the functionality of the corresponding commands that the LaTeX babel package provides. Naturally, our aim was to provide an "established" interface and not to re-implement the babel package.

Last but not least, the command `\InuitToday` is the Inuktitut version of the `\today` command.

4 The implementation details

We believe that good software should always have good documentation. Apart from creating TeX and METAFONT, Donald E. Knuth created the so-called *literate programming* methodology for program development. Roughly, this methodology is based on the observation that when one describes what he wants his program to do, he can implement it more easily. This program methodology is an offspring of the structured programming methodology of the 1960's. Although nowadays there are many new program development methodologies (e.g., generative programming), we still believe literate programming is quite adequate for the development of LaTeX/Λ packages. So we decided to implement our package using the literate programming tools for LaTeX (i.e., the `doc` package and the `docstrip.tex` TeX program originally developed by Frank Mittelbach).

Another decision we had to make was which character set to use. Naturally, since the "default" character set for Ω is the UCS-2 character set, one may opt to use this set. However, UCS-2 encoded files cannot be viewed "out of the box" on most computer platforms. Practically, this means that one should stick to good old ASCII even when developing Λ packages. Of course, many readers will object to this idea, but for the moment I believe this is the best option for development of packages for Λ.

Now we proceed with the various implementation details. In what follows we assume familiarity with ΩTPs. Readers not familiar with ΩTPs should consult the Ω documentation (e.g., see [2]).

4.1 The macros

By default, Λ uses the UC font encoding for monospaced fonts. However, it is quite possible that the user has a Λ format built without the necessary patch, so we first need to make sure that UC is the default font encoding for monospaced fonts:

```
\IfFileExists{ot1uctt.fd}{%
   \def\ttdefault{uctt}}{}%
```

As we noted in the previous section, the oinuit package offers a number of options. Here we describe how we implemented this facility. We present the relevant code for only two options for reasons of brevity:

```
\DeclareOption{nunavut}{%
   \ocp\InInuit=Qinuit2uni
   \ocplist\InInuitList=
       \addbeforeocplist 1 \InInuit
      \nullocplist
}
%
  ..........................
%
\DeclareOption{ucs2}{%
   \ocp\InInuit=id
   \ocplist\InInuitList=
       \addbeforeocplist 1 \InInuit
      \nullocplist
}
%
```

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
20	SPC	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	‘	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
80	Δ	Δ̇	ᐅ	ᐅ̇	ᐊ	...	ᐊ̇	Λ	Λ̇	ᐳ	ᐳ̇	ᐸ	ᐸ̇	ᐸ̇	Ո	Ո̇
90	Ɔ	‘	’	“	”	•	-	—	Ɔ̇	™	ᑕ	ᑕ̇	ᑦ	ᑭ	ᑭ̇	ᑯ
a0	NB SP	ᑯ̇	ᑲ	ᑲ̇	ᑳ	ᑎ	ᑎ̇	ᒍ	ᒍ̇	©	ᓗ	ᓗ̇	ᓘ	ᒥ	®	ᒧ̇
b0		ᒪ	ᒪ̇	ᒪ	ᒪ̇	ᒫ		ᓂ	ᓄ	ᓅ	ᓇ	ᓈ	ᓐ	ᓯ	ᓭ	ᓭ̇
c0	ᓴ	ᓴ̇	ᔆ	ᔆ̇	ᔇ	ᔨ	ᔩ	ᔪ	ᔫ	ᔭ	ᔮ	ᔾ	ᕆ	ᕇ	ᕈ	ᕉ
d0	ᕒ	ᕒ̇		ᕕ	ᕕ̇	ᕗ	ᕗ̇	ᕙ	ᕙ̇	ᕚ	ᕚ̇	ᕝ	ᕝ̇	ᕞ	ᕞ̇	ᕠ	ᕠ̇
e0	ᕡ	ᕡ̇	ᕢ	ᕢ̇	ᕣ	ᕣ̇	ᕤ	ᕤ̇	ᕥ	ᕥ̇	ᕦ	ᕦ̇	ᕧ	ᕧ̇	ᕨ	ᕨ̇
f0	ᕩ	ᕩ̇	ᕪ	ᕪ̇	ᕫ	ᕫ̇	ᕬ	ᕭ	ᕭ̇	ᕮ	ᕮ̇	ᕯ	ᕯ̇	ᕰ		

Table 2: The Inuit character table that the `inuitscii` ΩTP implements.

```
\ExecuteOptions{nunavut}
\ProcessOptions
```

Each option corresponds to some input encoding. Thus we declare an ΩCP list that can be pushed on the ΩCP stack to become the default input method. Note that since UCS-2 is the default input encoding, we need to leave the input intact when this option is used. This is the reason we load the `id` ΩCP.

Before we proceed with the definition of a number of commands, we need to input the font encoding file.

```
\input litenc.def
```

The encoding file contains only absolutely necessary information:

```
\DeclareFontEncoding{LIT}{}{}
\DeclareFontSubstitution{LIT}{cmr}{m}{n}
\DeclareErrorFont{LIT}{cmr}{m}{n}{10}
```

In the case of the Λ format being built with more than one set of hyphenation patterns, we need a command that can be used to select the hyphenation patterns we wish. We opted to implement a command that is available with the babel package, so that users are familiar with its use:

```
\def\selectlanguage#1{%
 \expandafter
 \ifx\csname l@#1\endcsname\relax%
   \typeout{^^J Error:No hyphenation
     patterns for language #1 loaded,}%
    \typeout{ default hyphenation
              patterns are used.^^J}%
  \language=0%
  \else\language=\csname l@#1\endcsname%
  \fi}
```

The declaration `\inuittext` should be used to permanently change the font encoding and pop the corresponding ΩCP list:

```
\def\inuittext{%
     \fontencoding{LIT}\selectfont%
    \def\encodingdefault{LIT}%
    \selectlanguage{inuit}%
    \pushocplist\InInuitList%
}
```

Now it is easy to implement a new environment that has the same functionality. We invite the reader to try to implement this new environment and to compare his/her implementation with ours.

As we have already explained, the command `\InuitToday` prints the current date in the Inuktitut language. Here is the code:

```
\DeclareRobustCommand{\InuitToday}{{%
   \fontencoding{LIT}\selectfont%
   \number\day\space%
   \ifcase\month%
    \or ^^^^1528^^^^14c4^^^^140a^^^^1546
    \or ^^^^1555^^^^1433^^^^140a^^^^1546
    ..............................
   \fi%
   \number\year}}
```

Note that the Inuktitut letters are typed in using Ω's `^^^^hhhh` notation, where `hhhh` are lowercase hexadecimal digits. With this notation we can specify the code point of any UCS-2 Unicode character, analogous to TEX's `^^hh` notation.

4.2 The Ω Translation Processes

To provide the functionality described above, we designed three ΩTPs: `inuitscii`, `Ninuit2uni` and `Qinuit2uni`. The first of these implements the 8-bit codepage presented in Table 2, while the other two allow users to enter Inuktitut text using the Latin transcription presented in Table 1. The `Ninuit2uni` ΩTP produces Inuktitut text that follows the Anglican orthography, while the `Qinuit2uni` ΩTP produces text that follows the Catholic orthography.

Because `Ninuit2uni` produces a character set that is a superset of the one `Qinuit2uni` produces, we describe only the structure of the first ΩTP. To begin with, we present the input and output sections:

```
input : 1;
output: 2;
```

Ω reads single byte characters and produces two-byte characters. The first thing we must handle in the expressions section is the vowels of the syllabary:

```
expressions:
   ‘i’ ‘i’ => @"1404;
   ‘i’     => @"1403;
   ‘u’ ‘u’ => @"1406;
   ‘u’     => @"1405;
   ‘a’ ‘a’ => @"140B;
   ‘a’     => @"140A;
   ‘h’     => @"157C;
```

Here we see that two consecutive vowels are mapped to one character, which is actually the dotted version of the character that the single vowel is mapped to. Note that if we change the order and try to handle the short vowel first, it will not be possible to handle the long vowel. So one must be very careful when designing ΩTPs.

Now, we will describe how we handle syllables that start with a particular consonant. If the consonant is the last character of the input stream, we simply push its Unicode equivalent to the output stream. If the consonant is not followed by one of the vowels, then we push the character that immediately follows the consonant back to the input stream and push the corresponding Unicode character to the output stream. Finally, depending on the vowel (or vowels) that follow the consonant, we push to the output stream the Unicode character that corresponds to this syllable. Here we show only the case for the consonant *p*:

```
‘p’ end:           => @"1449;
 ‘p’ ^(‘i’|‘a’|‘u’) => @"1449 <= \2;
‘p’ ‘i’ ‘i’        => @"1432;
‘p’ ‘i’            => @"1431;
‘p’ ‘a’ ‘a’        => @"1439;
‘p’ ‘a’            => @"1438;
‘p’ ‘u’ ‘u’        => @"1434;
‘p’ ‘u’            => @"1433;
```

The other consonants are treated the same way.

The `inuitscii` ΩTP is programmed in a different way. Here we are dealing with an 8-bit code page (namely ISCII) that is essentially an extended ASCII character set. This means that the lower part of the character set will be identical to ASCII and the upper part will contain the Inuktitut letters. So we define an array whose elements are the code points of the Inuktitut characters:

```
tabInuitSCII[@"81] = {
@"1403, @"1404, @"1405, ........
@"1432, @"1433, @"1434, ........
.............................
 @"1672, @"1673, @"1674, ........};
```

Now, we need a way to map the Inuktitut letters to the corresponding Unicode characters. The mapping is not difficult:[1]

```
@"00-@"7F  => \1;
 @"80-@"FF  => #(tabInuitSCII[\1-@"80]);
.           => @"FFFD;
```

Characters that belong to the lower part of the ISCII are mapped to themselves. Characters that belong to the upper part of the ISCII are mapped to a table entry that is located at $c-128$ where c is the ordering number of the Inuktitut letter in the ISCII character set. Note that the hexadecimal number 80 is equal to the decimal 128.

4.3 The fonts

The oinuit package uses virtual fonts that are built around the Computer Modern sans serif font and a PostScript version of the Nunacom TrueType font developed by Nortext (`http://www.nortext.com`), which is redistributed with permission from Nortext. We use the Computer Modern sans serif font because this better matches the Nunacom font we use.

Here we present only the general structure of the Ω virtual property list files that we had to create to allow users to enter UCS-2 encoded text directly. In the beginning of each ΩVP file, we have the identification part and the assigment of the various font dimensions:

```
(FAMILY OINUIT)
 (CODINGSCHEME Unicode Inuit)
(DESIGNSIZE R 10.0)
(FONTDIMEN
   (SLANT R 0.0)
```

[1] After all, in computer science arrays sometimes are treated as functions or, more generally, as mappings.

```
(SPACE R 0.5)
(STRETCH R 0.3)
(SHRINK R 0.1)
(XHEIGHT R 0.583)
(QUAD R 1.0)
)
```

Each virtual font includes the ASCII characters and the characters used in Inuktitut. So the font dimensions are really "average" font dimensions. The two different fonts are introduced with `MAPFONT` definitions:

```
(MAPFONT D 0
   (FONTNAME Inuit)
    (FONTDSIZE R 10.0)
   )
(MAPFONT D 1
   (FONTNAME cmss10)
    (FONTDSIZE R 10.0)
   )
```

Although character entries are quite standard, we present just one so that readers can see what has to be done.

```
(CHARACTER H 0021
   (CHARWD R 0.256)
   (CHARHT R 0.689)
   (CHARDP R 0.004)
   (MAP
       (SELECTFONT D 0)
      (SETCHAR 0 41)
      )
   )
```

It is important to note that we had to create the font `cmssbxo10` in order to have the matching Computer Modern sans serif bold oblique font for the corresponding Nunacom font.

4.4 Hyphenation patterns

Hyphnenating Inuktitut documents written in syllabics is fairly easy because there are no hyphenation rules! However, breakpoints cannot appear before any final consonant (or diacritic signs), except for bigger symbols — such as the symbols for *ng* or *q* — which include a final consonant within the symbol itself.

By default, all these Inuktitut letters have catcode "other", so we set it to "letter". In addition, we set the lowercase codes and the uppercase codes of each symbol. Since there are no uppercase or lowercase letters, we define that the uppercase/lowercase of a symbol is the symbol itself. Here is an example declaration:

```
\catcode‘^^^^1403=11
 \lccode‘^^^^1403=‘^^^^1403
 \uccode‘^^^^1403=‘^^^^1403
```

The fact that we can break a word at any point can be expressed as follows: Given a letter `c`, the pattern `c1` means that it is possible to break a word just after this letter. In addition, the exception is expressed as follows: Given a letter `c`, the pattern `2c` prohibits hyphenation before the letter `c`. Here is an "excerpt" from the patterns declarations:

```
\patterns{%
 ^^^^14031 ^^^^14041 ........
 ............................
 2^^^^1449. 2^^^^1466. ......
}
```

5 Conclusions and future work

We have presented our views regarding package development for Λ, and along these lines we presented the design principles of a particular package. We believe that our work can be used as a starting point for development of a set of widely-accepted principles for the development of Λ packages. This would be particularly useful in the framework of the LaTeX3 project. At any rate, we plan to use our experience to implement a number of other packages that will provide the TeX community with new typesetting capabilities.

6 Acknowledgments

I would like to thank Andrea Tomkins for giving me the right to redistribute the Nunacom font. I also thank Luis-Jacques Dorais, who explained to me the hyphenation rules of the Inuktitut language, and Dimitrios Filippou, who clarified the secrets of transforming these rules into hyphenation patterns suitable for use with TeX/Ω. Finally, thanks to the *TUGboat* reviewers Steve Peter and Karl Berry for clarifying the linguistics section and offering other general suggestions.

References

[1] Kenn Harper. Writing in Inuktitut: An Historical Perspective. Available from `http://www.nlc-bnc.ca/nord/h16-7301-e.html`, September 1983.

[2] Apostolos Syropoulos, Antonis Tsolomitis, and Nick Sofroniou. *Digital Typography Using LaTeX*. Springer-Verlag, New York, N.Y., USA, 2003.

◇ Apostolos Syropoulos
366, 28th October Str.
GR-671 00 Xanthi, Greece
`apostolo@ocean1.ee.duth.gr`