Bibliography

Standards

X/Open, “Portable Layout Services: Context-dependent and Directional Text,” Snapshot version, 1994

General Books and Articles

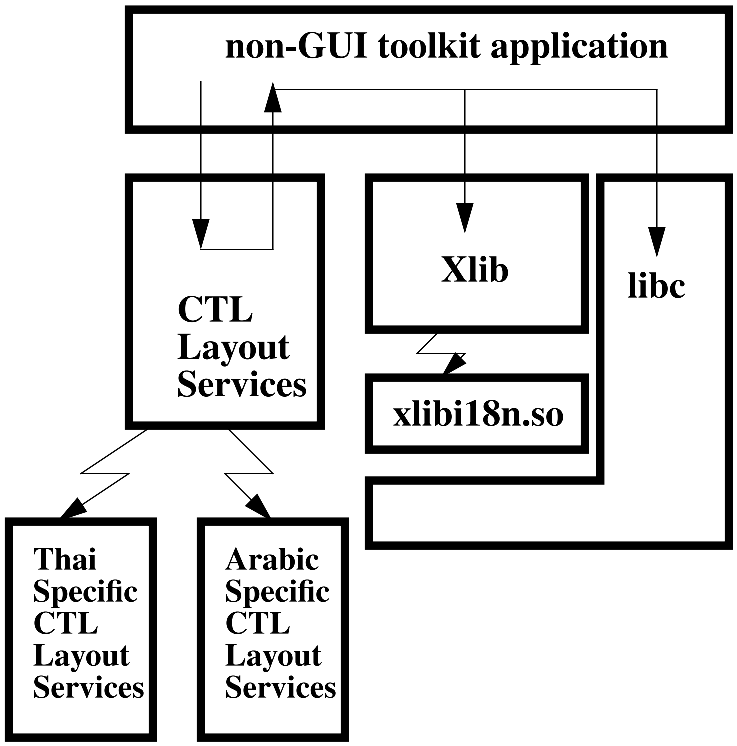
Akira Nakanishi, “Writing Systems of the World,” Charles E. Tuttle Company Inc., 1991

The Unicode Consortium, “Unicode / ISO 10646 Implementers’ Workshop 5 Tutorial,” The Unicode Consortium, 1994

Joseph D. Becker, “Arabic Word Processing,” Communications of the ACM, July 1987

Murat Tayli and Abdulla I. Al-Salamah, “Building Bilingual Microcomputer Systems,” Communications of the ACM, May 1990

5.2 Non-GUI Toolkit based application in X environment



Technical Implication.

- Non-GUI toolkit applications modification necessary.
- Internationalized applications link with additional library.

6.0 Layout services specification

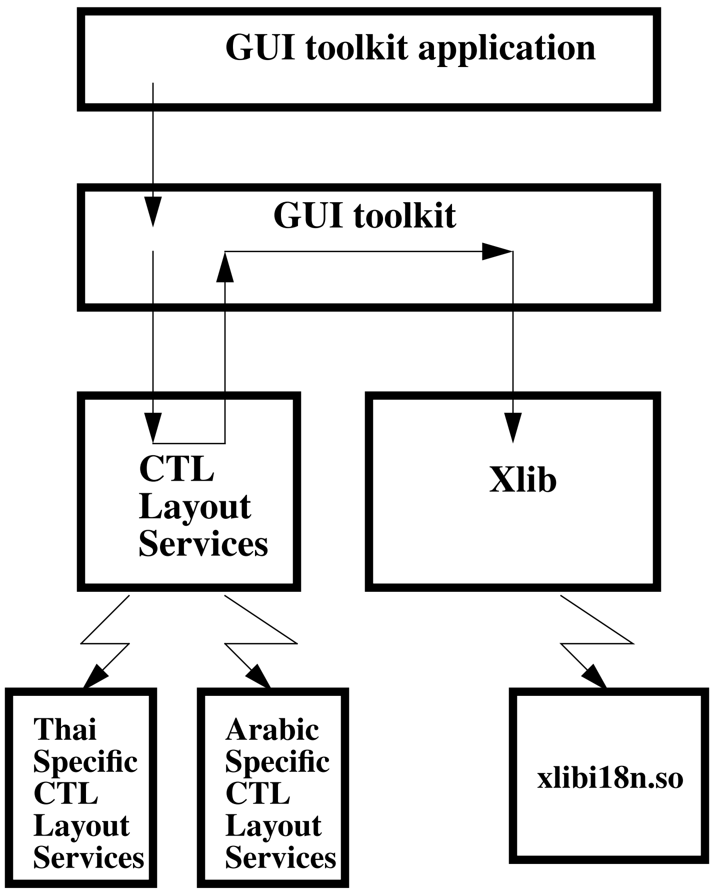
Work is underway with X/Open to standardize the interface layer (APIs) to the (CTL) Layout Services module. The interface is documented in the X/Open Layout Services specification. A snapshot version of this specification was published by X/Open in December 1994. The specification also includes description of the interface to the locale specific (CTL) Layout Services module as an appendix, although that is not considered to be part of the standard. The LS architecture is platform independent and can be applicable to any application including WWW browsers that requires the text layout services to enable languages such as Thai or Arabic.

Sun Inc, who is one of the sponsors of this specification, has completed a sample implementation of the LS framework to enable Thai and Arabic for Motif-based applications.

5.0 Layout services Architecture

The Layout services (LS) framework consists of a locale independent module that can load a locale specific module. The following two diagrams illustrate the architecture of two internationalized applications in the X environment, one which is GUI toolkit based and the other a non-GUI toolkit based. Both applications are linked with the (CTL) Layout Services[1] library which wil load a locale specific (dynamic loadable) module during application start-up. If the locale specific module is not available, no text layout will be performed prior to rendering.

5.1 GUI Toolkit based application running in the X environment



Technical Implication

- No source code change to GUI toolkit applications.
- Toolkit modification necessary.
- Internationalized applications link with additional library.

1. CTL Layout Services is responsible for character shaping & composition, national numbers mapping and bidirectional text layout. Please refer to section 1.1 to 1.3 for more information of these functionality.

While implicit text-type is the more desirable approach, an Arabic output method may need to support all three types for backward compatibility reasons.

Unicode supports a method called the Basic Display Algorithm, which is a combination of implicit and explicit text-type. This is done by introducing a few specific directional control characters.

4.4 Symmetrical Swapping

Symmetrical Swapping is the transformation of text segment which includes character with implied directional meaning. For example in Arabic, the logical order of a > b in storage should be transformed into b < a when it is rendered. Other characters that require symmetrical swapping are parentheses, square brackets, braces etc.

In addition, there are two types of ordering associate with bidirectional text processing.

Logical order: The order which the text is input by the users and processed by applications.

Physical order: The order which the text is rendered on the screen.

For example, if “home” is a word of a right-to-left script, the user will enter “h”, “o”,”m” and “e”. The logical order “home” will be stored internally. When this word is displayed, a layout services will reorder it to the physical order of “emoh”.

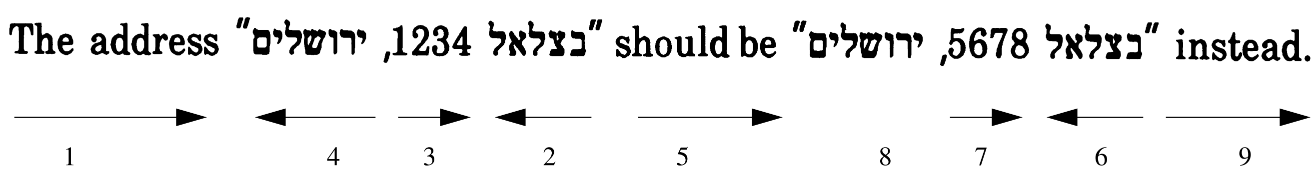
b. Global Orientation

c. Text-Types

d. Symmetrical Swapping

4.1 Directional Segment

Directional Segment is a portion of the sentence with a different directionality. This determines the directional flow of the text when a sentence contains scripts of mixed direction. A segment of right-to-left direction could be embedded or "nested" within it an additional segment of left-to-right direction. In theory, there could be many levels of nesting. However, only two levels are needed in most cases.



This is an example of a bi-directional text sentence of writing two Hebrew addresses in an English sentence. The sentence is divided into nine segments. The number of each segment indicates the order of the directional flow.
The English in segment 1, 5 and 9 have a nesting level of 0. The Hebrew text in segment 2, 4, 6 and 8 have a nesting level of 1. Since the numerals in segment 3 and 7 are nested within the Hebrew text, they have a nesting level of 2.

4.2 Global Orientation

Global Orientation is the predominant direction of the text. It is used to determine directional flow of the sentence and the movement of the caret in a mixed direction sentence.

4.3 Text-Types

Text-Types define the format of Arabic text to be stored on disk:

visual text-type is for storing text in physical order. No layout is needed for rendering. However, transformation is needed for text processing. Many legacy environments use this method[1].

implicit text-type is for storing text in logical order. The directionality of the sentence is determined by the inherent characteristics of the segment. The X/Open proposal mainly proposed this method.

explicit text-type is for storing text in logical order. The directionality of the sentence is determined by explicit control characters.

1. Need further investigation for the level of visual text-type support.

가 누 듸

Example 1 *Example 2* *Example 3*

륀 믈

Example 4 *Example 5*

Figure 10 Korean Hangul examples of the five different placement in figure 8

This shaping and character composition process should be done automatically. However, the user should be able to deactivate this feature when it is not desirable.

3.0 National Numbers

The Arabic script can be written using both "Arabic" digits as in Latin script, and "Hindi" digits. In an Arabic system, it is common that digits are stored in the encoding of "Arabic" digits. Depending on the usage, the software could convert to "Hindi" digits during rendering. Usually, "Arabic" digits are used for data processing application while "Hindi" digits are used for word processing application.

٠ ١ ٢ ٣ ٤ ٥ ٦ ٧ ٨ ٩	Hindi digits
0 1 2 3 4 5 6 7 8 9	Arabic digits

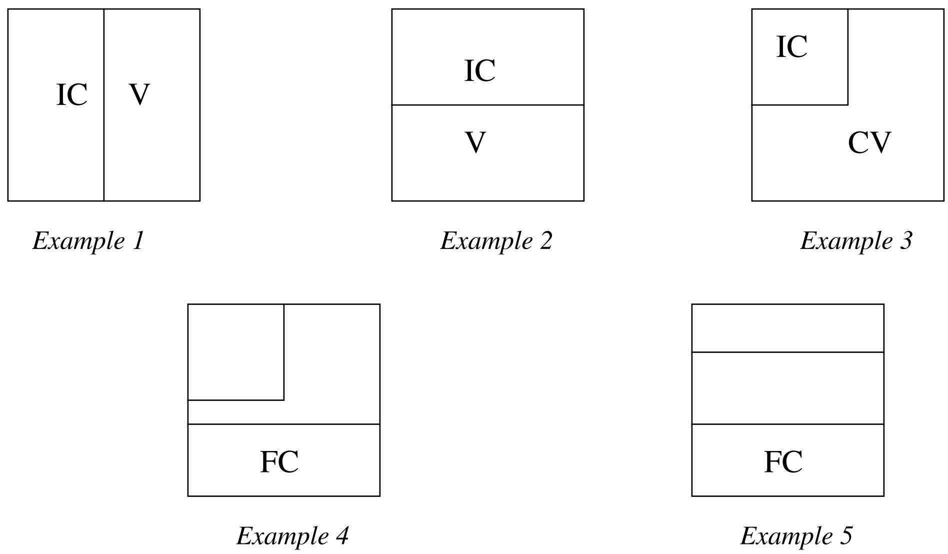
Figure 11 Hindi digits (used in Arabic)

4.0 Bidirectionality

Bidirectionality occurs when a left-to-right script is intermixed with a right-to-left script. For instance, when Arabic and Latin script are intermixed the following issues has to be addressed when dealing with mix directionality.

a. Directional Segment

- For combined vowel which has both horizontal and vertical representation, the initial consonant is displayed at the upper-left corner, and the combined vowel is at the right side underneath the initial consonant. *(Example 3)*
- The optional final consonant is always displayed at the bottom portion of the display cell. *(Example 4 and 5)*



IC: Initial Consonant
V: Vowel
CV: Combined Vowel
FC: Final Consonant

Figure 9 Five possible placement of displaying Korean Hangul characters

Three Component Display Cell   กู่ รี่ ปี่

Floating Tone Mark   รี่ ร่ กั กี้

Sara Am   ำ กำ ป่ำ ฝ๋ำ

Figure 7 Example of Thai text

2.3 Korean Hangul script:

Korean Hangul alphabet has 14 base consonants and 10 base vowels.

ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅅ ㅇ ㅈ ㅊ ㅋ ㅌ ㅍ ㅎ

ㅏ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ

Figure 8 14 basic consonants and 10 basic vowels of Hangul alphabet

The Korean Hangul script has following characteristics.

a. Hangul grammar, two or more base consonants and base vowels can be combined to form a consonant and a vowel, respectively. There are about 27 such combined elements.

b. Each Hangul syllable has following composition elements:
   initial consonant, middle vowel and optional final consonant.

c. Four possible combinations of such elements can be displayed in a single cell.
   Consonant
   Vowel
   Consonant + Vowel
   Consonant + Vowel + Consonant

d. Each composition element can be displayed in various positions within a display cell depending on the number of elements in the display cell and the shape of each element. The following are the basic rules for composition.
   - A consonant or a vowel always positions at the center of the display cell, if it is the only element
   - The initial consonant can be displayed at the left half, the upper half of the display cell depending on the shape of the vowel. *(Example 1 and 2)*
   - The vowel can be displayed on the right half or lower half of the display cell. *(Example 1 and 2)*

The Thai script has the following characteristics.

a. Four types of characters:
   vowels, consonants, tone marks and diacritical marks.

b. Up to three characters can be displayed in a single display cell.

c. Each display cell is divided into 4 rows of display areas for different types of characters:

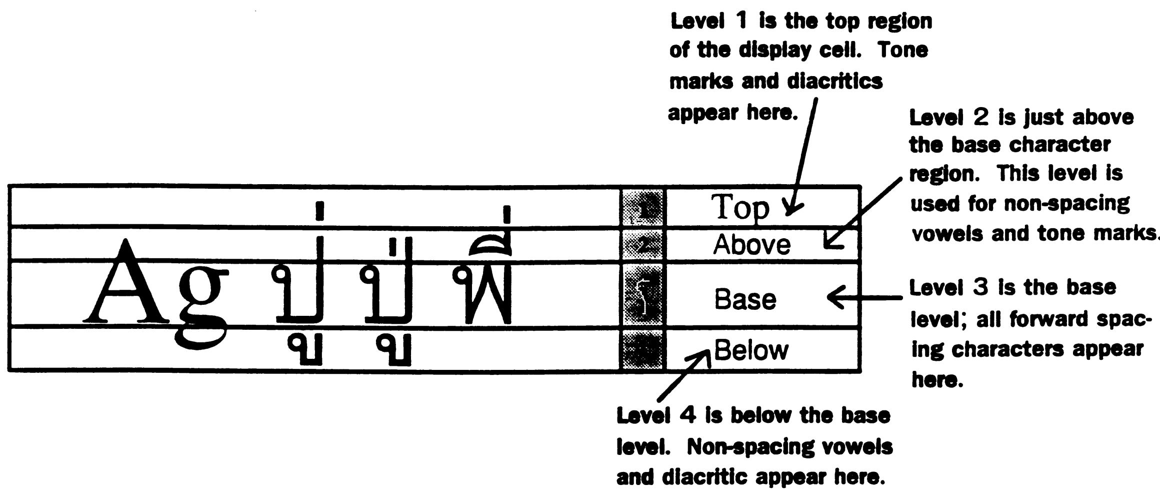


Figure 6 Thai text display cell diagram[1]

Note: If a tone mark character follows a vowel character, the tone mark character will appear in Row 1. If a tone mark character follows a consonant character, it will appear in Row 2.

d. Eight possible combination of characters to be displayed in a single cell.

   Consonant + Vowel + Tone mark

   Consonant + Tone mark

   Consonant + Vowel

   Consonant + Diacritical mark

   Diacritical mark

   Vowel

   Vowel + Tone mark

   Tone mark

1. Diagram taken from "Computer with Thai Language", ISBN974-7570-66-1

b. Multiple characters can be combined into a single ligature glyph.

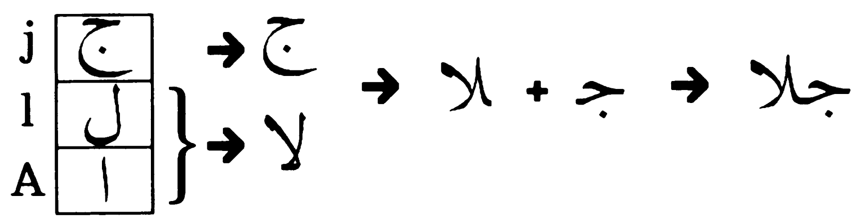


Figure 4 Ligature lA joining in the word jlA (jalla, meaning “it cleared”)

c. Has non-spacing diacritics.[1]

2.2 Thai script:

Thai script has 44 base consonants and 11 base vowels.

Figure 5 The base consonants, vowels and numerals of Thai script.

1. Diacritics in Arabic are like vowels in English. Diacritics should be rendered as non-spacing marks, which means rendering the diacritics character does not change the current display position. However, due to font limitation, some implementations only support spacing diacritics.

The Arabic script has the following characteristics.

a. Some characters may be displayed in different shapes depending on the placement of the character within a word.

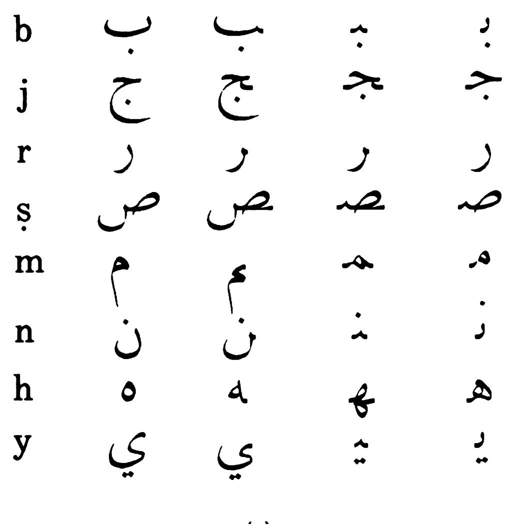


Figure 2 Each row shows 4 possible shapes of the character (Isolated, Initial, Middle and Fin*al)*

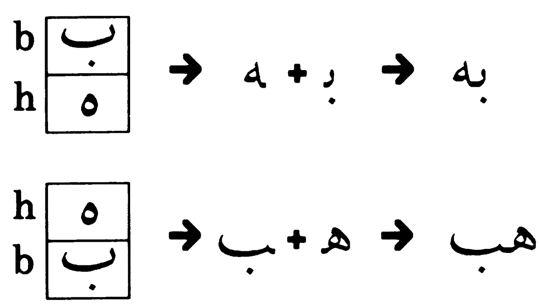


Figure 3 The Arabic letter “b” and “h” assume a different form depending on their placement

2.1 Arabic script:

Arabic script has 28 letters and is written from right to left.

ا	ب	ت	ث	ج	ح	خ
A	b	t	th	j	ḥ	kh
د	ذ	ر	ز	س	ش	ص
d	ð	r	z	s	sh	ṣ
ض	ط	ظ	ع	غ	ف	ق
ḍ	ṭ	ð̣	ʿ	ġ	f	q
ك	ل	م	ن	ه	و	ي
k	l	m	n	h	w	y

Figure 1 28 consonantal letters of the Arabic alpha*bet*

1.0 Complex Text Languages Overview

Complex Text Languages(CTL) refer to a group of languages whose scripts have one or more of the following characteristics:

context sensitive:	Characters can assume different shapes depending on neighboring characters and their placement within a word. This is the characteristic of Arabic, Thai, Korean Hangul and Urdu.
composite character:	Characters are composed by a few base elements. This is the characteristic of Thai, Lao, Vietnamese and Korean Hangul.
bidirectional:	Characters are written from right-to-left horizontally with exception of numbers which are written from left-to-right or when intermixing with other left-to-right scripts such as Latin. This is the characteristic of Arabic, Hebrew, Urdu, Farsi and Yiddish.

In most computer systems, the CTL text strings are stored in a different order and form than the displayed text due to text processing efficiency. Therefore, text strings of these languages need to be preprocessed prior to being displayed.

A locale specific "Layout Services" module is needed to support the displaying of CTL text strings. It should provide a combination of the following features:

1. Shaping & Character Composition
2. National Numbers
3. Bidirectionality

2.0 Shaping & Character Composition

Shaping is the process of displaying the appropriate form of a character in context sensitive scripts. Character composition is the process of displaying a composite character by combining a few basic elements. Arabic, Thai and Korean Hangul are good examples of languages which requires such processing due to their characteristics.

Layout Services for Complex Text Languages

Nelson Ng

Global Products Engineering, SunSoft Inc.

Introduction

Over the history of civilization, thousands of languages and hundreds of scripts have been developed as a vehicle for verbal and written communication. In today's world about twenty-nine scripts are still commonly used. Each script has similarities and differences. SunSoft currently supports Latin scripts (such as French, German, Italian, Swedish, Spanish) and Asian ideographic scripts (such as Chinese Hanji, Japanese Kanji and Korean Hanja). Although these two types of scripts look dramatically different, they share similar characteristics when being displayed by a computer. They both can be displayed in a single writing direction, for instance, left-to-right; require no special processing prior to rendering each character. However, a new rendering approach is necessary to support writing systems like Arabic, a cursive right-to-left script with characters which can be rendered in different shapes depending on their placement. Another example is Thai where multiple characters appear in a single display cell.

The special processing of characters prior to rendering is collectively referred to in this document as “Layout services”. The goal of this document is to describe an architecture for applications using Layout Services to enable lanaguages such as Arabic and Thai. We will refer to such languages as Complex Text Languages (CTL). The Layout Services architecture is platform independent and can be applied to applications on UNIX, Windows, Macintosh and WWW.

This document will:

(i) Provide a definition of CTL

(ii) Define the goals for supporting CTL

(iii) Propose an architecture for CTL support using Layout services