

Arabic Script SHOULD NOT be so Scary!

What else Unicode still needs to do
for the Perso-Arabic script

Behnam Esfahbod
behnam.es

In the past decade...

The FarsiWeb Project

- National Standard for Persian Unicode text
- Standard Persian keyboard layouts on all major OSs
- Standard-based Open Source Persian TTF fonts
- Persian support in GNOME & Mozilla projects

IRNIC, The .IR ccTLD Registry

- Deploy Persian (second-level) IDNs
- Apply for the IDN ccTLD (actually, 2 of them!)

Persian Computing Community

- Over 300 engineers, designers,
- SIGs, e.g. Persian Typography

ICANN & IETF

- IDN Variant Issues Project
- MESWG Task Force on Arabic-Script IDNs

Previously on Unicode

Perso-Arabic Script

Mainly Arabic and Indo-Iranian languages

- Is the basis for many alphabets
- Persian, Urdu, Pashto, ..., and Arabic!

No alphabet uses all kinds of shapes

- Four-dots is not used in Arabic or Persian
- But 99% of the shapes are recognized by everyone

Many writing styles

- Naskh, Nasta'liq, etc

Not all alphabets use all the features

- Arabic language/alphabet doesn't use ZWNJ, traditionally
Routed in the language properties and its conjugation forms
- But recently is being used for non-Arabic words

IBM ⇒ آی بی ام

Semantic Encoding

Writing Direction

- Letters: right-to-left
- Digits: left-to-right

⇒ Unicode Bidirectional Algorithm (Bidi/UBA)

س ل ا م

One code-point for each letter

⇒ Unicode Arabic Joining Algorithm

سلام

سلام

- Joining Control Characters

0600

Unicode

06FF

Arabic																
	060	061	062	063	064	065	066	067	068	069	06A	06B	06C	06D	06E	06F
0	◌ْ	◌َ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ
1	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ
2	◌ْ	◌َ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ
3	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ
4	◌ْ	◌َ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ
5	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ
6	◌ْ	◌َ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ
7	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ
8	◌ْ	◌َ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ
9	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ
A	◌ْ	◌َ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ
B	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ
C	◌ْ	◌َ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ
D	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ
E	◌ْ	◌َ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ	◌ِ
F	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ	◌ُ

0700

Unicode

07FF

Syriac					Arabic Supplement			Thaana				Nko				
	070	071	072	073	074	075	076	077	078	079	07A	07B	07C	07D	07E	07F
0						◌ْ	◌َ	◌ِ								
1						◌ُ	◌ُ	◌ُ								
2						◌ْ	◌َ	◌ِ								
3						◌ُ	◌ُ	◌ُ								
4						◌ْ	◌َ	◌ِ								
5						◌ُ	◌ُ	◌ُ								
6						◌ْ	◌َ	◌ِ								
7						◌ُ	◌ُ	◌ُ								
8						◌ْ	◌َ	◌ِ								
9						◌ُ	◌ُ	◌ُ								
A						◌ْ	◌َ	◌ِ								
B						◌ُ	◌ُ	◌ُ								
C						◌ْ	◌َ	◌ِ								
D						◌ُ	◌ُ	◌ُ								
E						◌ْ	◌َ	◌ِ								
F						◌ُ	◌ُ	◌ُ								

Dual-Joining (yellow) & Right-Joining (blue)

Special Characters

1. U+0640 ARABIC TATWEEL

- a.k.a. Kashida
- کشیده ⇒ ده_____کشید

2. U+200C ZERO WIDTH NON-JOINER

- a.k.a. ZWNJ
- نامهای ⇒ نامهای

3. U+200D ZERO WIDTH JOINER

- a.k.a. ZWJ
- ه.ش. ⇒ ه.ش.

Status Quo

1. Kashida

- On the keyboard: too similar to Hyphen
- Makes the text hard to process (search, ...)

2. ZWNJ

- Inaccessible in non-standard keyboard layouts
- Confuses search engines and applications
- IDNA2003
- UTR #31 (Unicode Identifier and Pattern Syntax)
- IDNA2008
- UTR #36 (Unicode Security Considerations)

3. ZWJ

- Too little use
But when it's needed, user is already tired of the other issues

Be Stylish

Typographic Styles

Western styles for Latin-like scripts

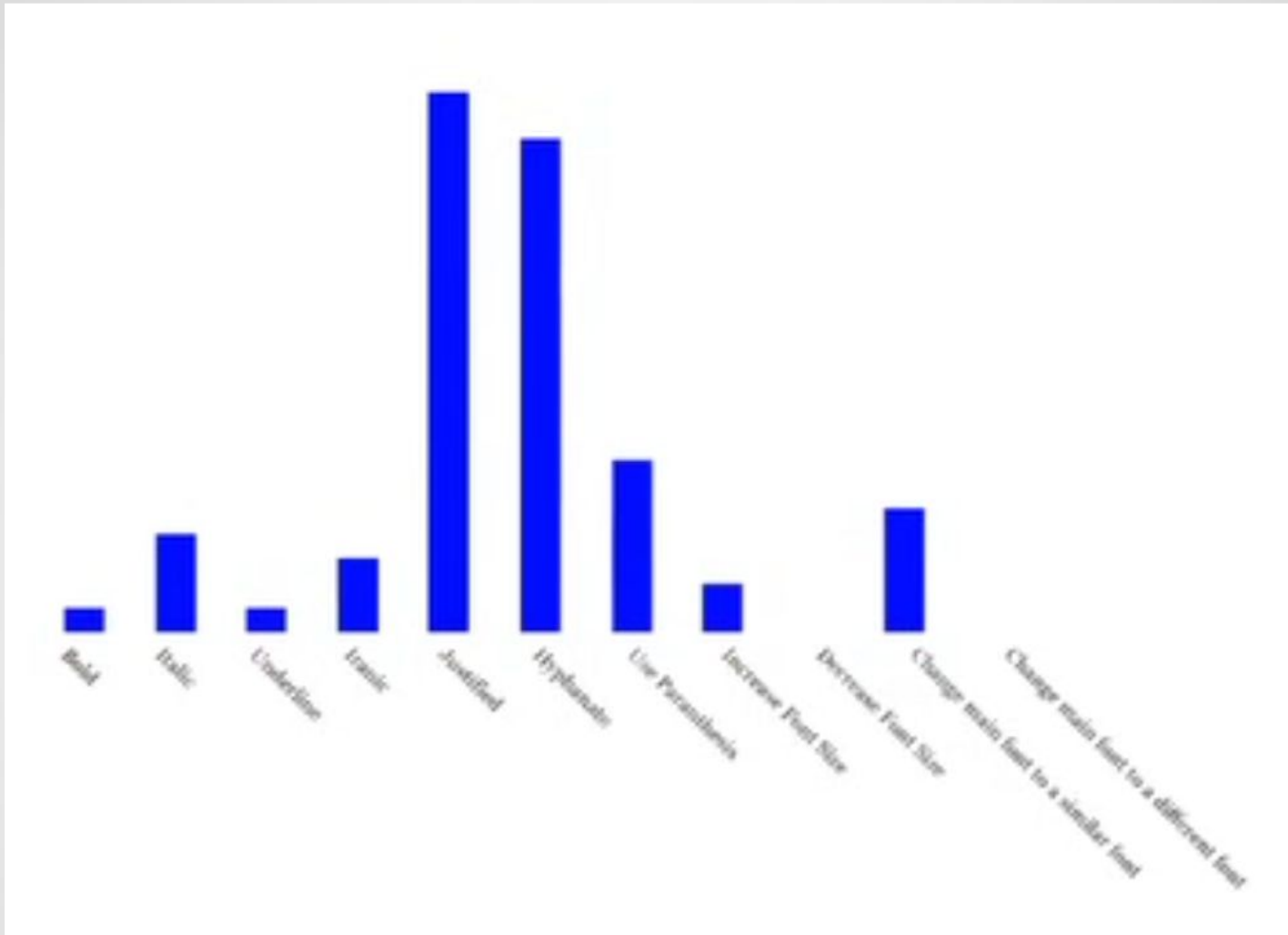
- Bold
- Italic & Slanted (including Iranic: left-slanted)
- Underline

Question:

When, how, and by whom these three became the de facto standard?

Justification & Elongation

- Traditionally used in manuscripts
- Used to be available in movable type too
- Recent study shows Elongation is still the right way to emphasize text in Perso-Arabic script



© 2013 Nasser Hadjloo, "Curious Case of Word Stylers", TEDxTehran

1000 persons × 10 emphasis methods
⇒ Justification & Elongation lead the way!

Typographic Emphasis

Letter Spacing

L A T I N S C R I P T

Elongation + Word Spacing

خط فارسی

- CSS
 - “letter-spacing” is useless (خط)
 - “text-justify” does work, only for justification
- ⇒ No sane way to do elongation

Verbal Emphasis

Letter Repetition

Latin script... Nooooo!

Elongation

خط فارسی... بلی!

- Keyboard

- Direct mapping of characters to typographic elements (glyphs) is bad!
- Press a Kashida key a few times to emphasize some letter, bad again!

Text Input & Storage

Joining Control

ZWNJ is very common in Persian

- Necessary to create compound words
- And mandatory for some words

خانه‌ای

بی‌بی

- But preferential in some others

خط‌کشی

می‌شود

Larry Tesler: “no modes”

- Mode: a distinct setting in which the same user input will produce perceived different results than it would in other settings

Modeless Input Methods

Current keyboards techniques

- Based on the typewriter technologies
- Based on the needs of Latin script

Kashida insertion should be implicit

- Three key presses give you three letters
- Three letters need three key presses

[ه].....[ل].....[ب]
ه ل ب

ZWNJ insertion should be implicit

- Instead of “mode”, we need a “modifier”
- Shift key would make much sense!

Note: only applies to dual-joining letters, if followed by a joining letter

Obstacles

Input

- Need to remove unnecessary ZWNJ dynamically
- Hard to implement in keyboard engines
- Have to be implemented in an IME

Processing

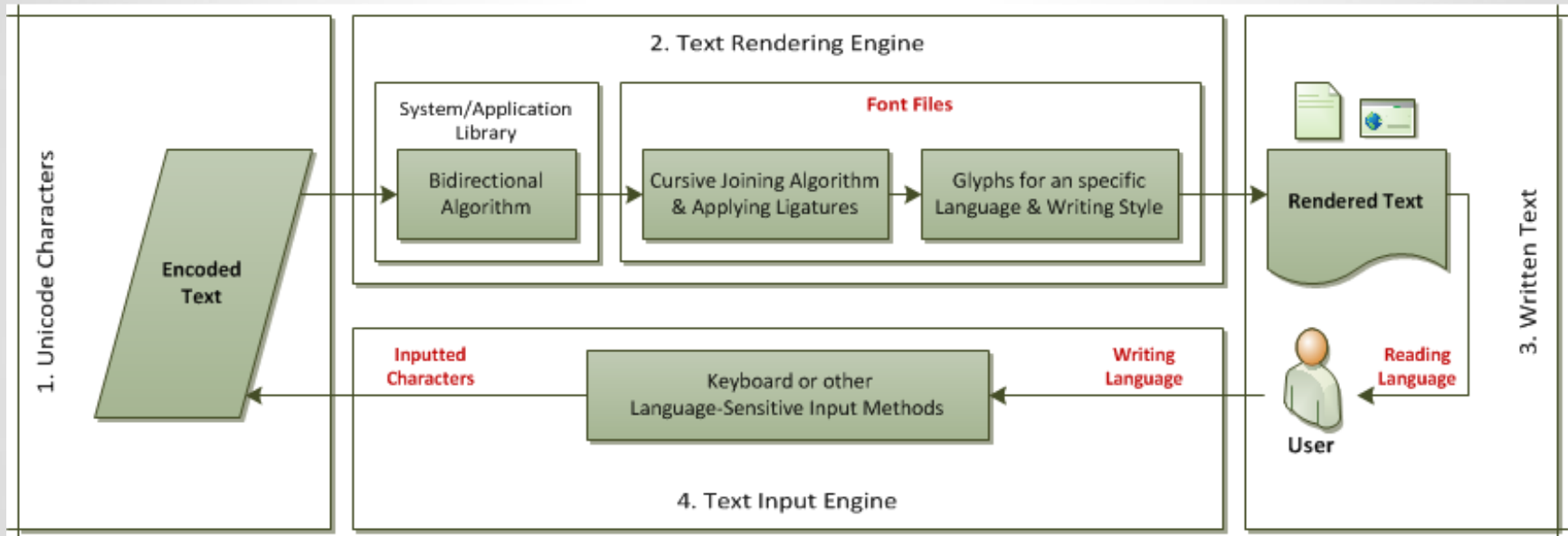
- Search
 - Kashida should be ignored
 - ZWNJ should be respected, sometimes
- Identifiers
 - Standards have to deal with them, individually!

Output

- A few apps still have problems with ZWNJ
- Kashida not handled correctly in most fonts

Multilingual Environments

HCI Model & Language



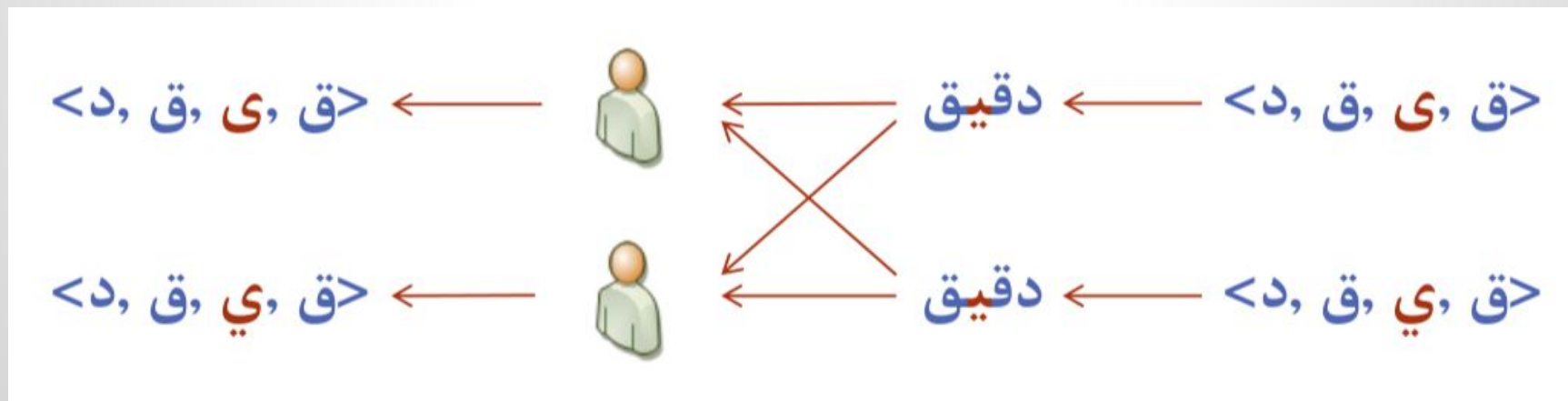
Language in the computer

Language in the user's mind

Encoding Challenges

Characters with similar shapes (Not in all joining forms)

Arabic Yeh vs. Persian Yeh



Typical default shapes for...

		<i>Isolate</i>	<i>Final</i>	<i>Medial</i>	<i>Initial</i>
U+0647	ARABIC LETTER HEH	ه	ه	ه or ه	ه
U+06BE	... DOACHASHMEE	ه	ه	ه	ه
U+06C1	... GOAL	ه	ه	ه	ه
U+06FF	... WITH INVERTED SMALL V ABOVE	ه	ه	ه	ه

Urdu

U+0647	ARABIC LETTER HEH	ه	ه	ه	ه
--------	-------------------	---	---	---	---

Sindhi

U+0647	ARABIC LETTER HEH	ه	ه or ه	ه or ه	ه
--------	-------------------	---	--------	--------	---

Parkari

U+0647	ARABIC LETTER HEH	ه	ه	ه	ه
--------	-------------------	---	---	---	---

Kurdish

U+0647	ARABIC LETTER HEH	ه	ه	ه	ه
--------	-------------------	---	---	---	---

© SIL International

Preferred shapes of letter Heh family based on languages

Internationalized Identifiers

Confused about Kashida

- Use NFKC to get rid of it

Confused about ZWNJ

- Some protocols mandate some rules
 - And FAIL if the input doesn't comply
 - UTR #31 (Unicode Identifier and Pattern Syntax) rule 2.3.A1
 - UTR #46 → IDNA2008 → UTR #31
- No standard approach to make it more user-friendly
 - Although the semantics are obvious to the user

No harmony in same-shape problems

- Protocols barely talk about it
- Policy-making works only for a few cases
- Applications decide how to handle it, if they do

Developing a Solution

Existing Similar Methods

Case-Mapping

- Letter cases (different representation of the same concept)
- Useful for western scripts (Latin, Cyrillic, Greek, ...)
 - Doesn't work for the other scripts
- Language-dependent

Normalization Form Canonical

- Deal with encoding issues
 - Ensure backward-compatibility
- Allow more expansion of UCD
 - Better forward-compatibility

Normalization Form Compatibility

- Works good for Arabic script (Kashida)
- But too much damage to other scripts

The Solution **SHALL** be able to...

- Remove unnecessary (invisible) ZWNJs
- Remove Kashida characters
- Place text into an specific language
 - Maintaining the expected shape
- Stay consistent when language is not specified

Arabic Shape Mapping

- Language-less Basic Normalization
 - Remove any unnecessary (invisible) ZWNJ
- Language-less Identifier Normalization
 - Basic normalization
 - Remove any Kashida
- Language-based Shape Mapping
 - Map characters with same joining-form shapes to the right one for the target language
- Language-less Shape Mapping
 - Not straightforward at all!

T H E E N D

پایان