Nuun: A System for Developing Platform and Browser independent Arabic Web Applications

Nizar Y.A. Habash
habash@umiacs.umd.edu

University of Maryland
Institute for Advanced Computer Studies
and NuunLabs

Abstract

For a human language to reach its full potential in cyberspace, platform and browser independent support for data entry and display are required. Arabic web applications are far from this state of ubiquitous support. Full support is available only under Arabic Windows, while little support is provided under other versions of Windows, and no support at all under UNIX systems .The Nuun toolkit addresses this challenge. Nuun uses an allographic encoding in which each letter form is encoded separately to provide an Arabic display capability in any web browser on any platform. The display capabilities are augmented with an input method that provides the necessary extensions to HTML forms to handle Arabic script. Nuun tools can be used to build interfaces for new or existing web applications that use any Arabic encoding, and can support any language based on the Arabic script (e.g., Persian, Urdu, or Kurdish).

Nuun: Un Système pour le Développement d'Applications Arabes sur le Web Independamment de la Plate-form et du Browser

Résumé

Pour qu'un langage humain puisse atteindre sa capacité totale dans le Cyberspace, il faut que la saisie et l'affichage de données soit indépendant de la plate-forme et du browser. Actuellement, les applications Web arabes n'ont pas de soutien omniprésent. Le toolkit Nuun a été conçu pour adresser ce défi. Il utilise un codage allographique dans lequel chaque forme du lettre est encodée à part. Ce codage fournit une capacité d'affichage pour les caractères arabes avec n'importe quel browser et sur n'importe quelle plate-forme. Cette capacité est augmentée avec une méthode d'entrée qui fournit les extensions nécessaires aux formes HTML pour la saisie des caractères arabes. Les outils de Nuun peuvent être utilisés dans la crèation d'interfaces pour des applications Web, soit nouvelles ou actuelles, basés sur n'importe quel codage arabe. Ils peuvent en outre soutenir n'importe quel langage utilisant les caractères arabes (par exemple le Persan, l'Ourdou, le Kurde etc.).

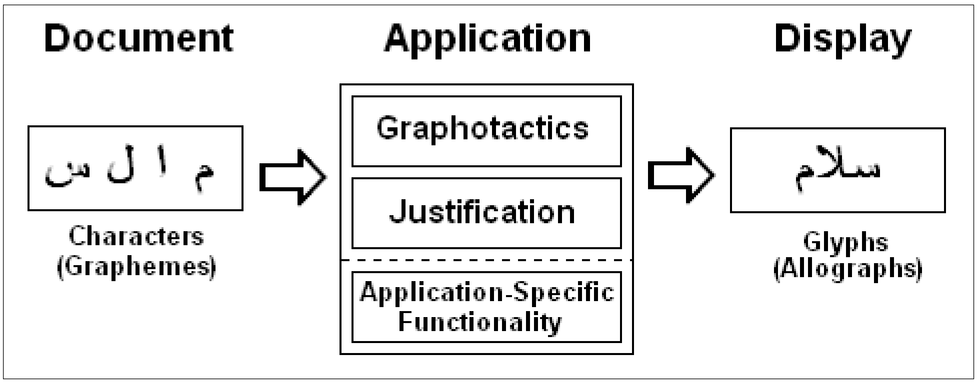
نون والقلم وما يسطرون.

Nuun. By the pen and that which they write, (Holy Quran 68:1)

Background

For a human language to reach its full potential in cyberspace, platform and browser independent support for data entry and display are required. Operating systems such as Arabic Windows coupled with browsers such as the Arabic edition of Internet Explorer provide the most comprehensive support for Arabic, but much of the world's computing infrastructure is not configured in this way. Browsers such as Tango can support Arabic on any version of the Windows operating system, but relatively few users will install such browsers when Netscape and Internet Explorer are both free. Windows 98, which includes support for displaying Arabic but not for Arabic data entry, is a step in the direction of ubiquitous worldwide support for Arabic. Unfortunately, it will be some time before legacy systems such as Windows 95 and the present version of Windows NT disappear, and the prospects for native support for Arabic computing on Unix platforms is still unclear.

While languages as diverse in their scripts as French and Chinese are supported by browsers on all platforms, Arabic support is behind, particularly because of two characteristics of Arabic script: right-to-left justification and many-to-many graphotactics. Arabic is written from right to left with lines flowing from top to bottom. So, Arabic text justification and line wrapping require special support. Arabic is different from other scripts, except Mongolian, in having complex graphotactics, rules that govern the surface form (allograph) of a letter (grapheme) based on the letter's environment. For example, the Arabic letter ع appears as one of four allographs: عـ (initially), ـعـ (medially), ـع (finally), and ع (stand-alone). Additionally, some letters combine together creating more complex graphemes such as ا + ل ➔ لا. More intricate Arabic writing styles require more complex combinations including vertical positioning of otherwise horizontally related characters.



Graph 1

All current standard Arabic encodings such as International Standards Organization ISO 8859-6, Arab Standards and Metrics Organization ASMO 449, Microsoft Arabic Windows Code Page 1256, and Unicode use the same type of graphemic encoding of Arabic. An Arabic string of text is encoded as a left-to-right sequence of characters corresponding to the letters constituting it, regardless of their surface forms. Justification, line wrapping and graphotactics (character to glyph mapping) are left to the individual application whether it be a browser or a word processor or something else (see Graph 1). Arabic data entry is as complicated as Arabic text display and its support is also left to the specific application.

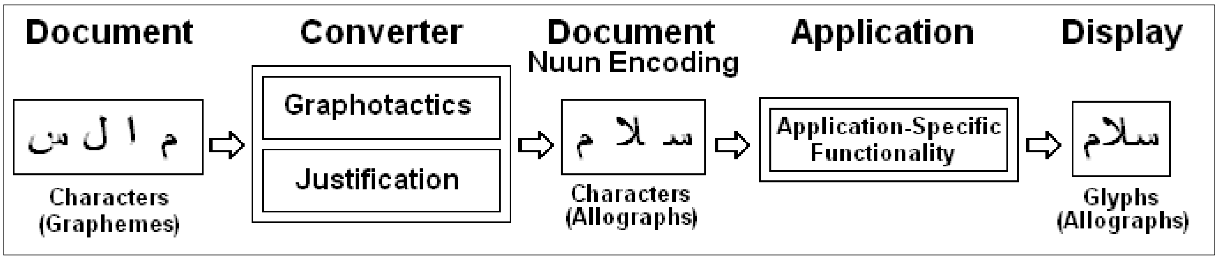
Although graphemic encodings of Arabic are the most economical, they are limited by their dependence on specific application support or platform support. Few people outside the Microsoft Arabic Windows' "bubble" in the Middle East are able or willing to install an Arabic enabled operating system or to download Arabic supporting browsers. To deal with the issue of platform and browser dependence, many solutions were developed to display Arabic documents on the web. The crudest and most common is to display Arabic documents as images. While this produces platform and browser independence, its disadvantages are numerous; ranging from expensive waste of time and memory space, to limitations on hyperlinking and searching documents. Another trend in Arabic web publishing is the use of PDF files. This method may allow for beautifully typeset Arabic documents but is not useful for interactive web applications and does not support running applets or Dynamic HTML within pages. It also requires the downloading of Acrobat Reader. Java applets for data entry and display have also been implemented taking advantage of Java's platform independence [Maeda et al.] and [www.maktoob.com]. Unfortunately, Java is relatively slow, and has limitations on displaying HTML documents in applets: the size of the applet would increase drastically with more features for displaying HTML that it becomes as expensive as downloading a browser every time one wants to read a page. Another solution is Allograph-to-Inline-Image, which uses an image for each Arabic allograph and typesets the document as a sequence of these images[www.ayna.com and www.postchi.com]. This solution requires different copies of the letters to format the text (bold, italic, different font sizes and colors). Also, adjusting the font size on the client machine is out of the question.

Solution

We have developed the Nuun toolkit as a platform and browser independent solution that bypasses the problems just discussed. Nuun uses an allographic encoding in which each letter form is encoded separately to provide Arabic display capability in any web browser on any platform. The display capabilities are augmented with an input method, NuunForm, that provides the necessary extension to HTML forms to handle Arabic script. Nuun tools can be used to build interfaces for new or existing web applications that use any Arabic encoding, and can support any language based on the Arabic script (e.g., Persian, Urdu, or Kurdish).

Nuun Encoding

Nuun encoding is quite different from the standard encodings described above. Arabic allographs are explicitly represented in the encoding as characters. Text strings are prejustified (right-left justification and line wrapping). The process of mapping characters to glyphs described above is broken into two parts, with a document whose characters are encoded allographically (Nuun encoding) as the link between these parts (see Graph2 and compare with Graph 1). First, text is mapped from a standard Arabic encoding into Nuun's prejustified allographic encoding, this step happens offline for web pages (it can also be created automatically in interactive web applications). This phase takes the complexity of graphotactics and justification from applications not designed to support Arabic. It technically reduces the problem of character to glyph mapping in Arabic to that in Roman script languages: from many-to-many to one-to-one.



Graph 2

One previous attempt to use allographic encoding of Arabic script [www.neda.net] resulted in less than satisfactory support because it limited itself to the use of the character encoding space in upper ANSI (128-255) following the tradition of reserving lower ANSI for ASCII encoding. This, of course, limits the number of letter forms that can be encoded. For example, in the attempt mentioned above, there is no support for Arabic vocalization diacritics (*tashkiil*). With Nuun, the whole encoding space (0-255) is used to encode Arabic allographs thus providing enough variant forms and diacritics to support documents requiring vocalization diacritics such as Quranic text.

Because the encoding of letters in Nuun is different from other encodings, a special font needs to be used to view Nuun-encoded documents. For web applications, this font can be downloaded and installed on the user’s machines or it can be embedded in the HTML document containing Nuun-encoded text. The two options will not interfere with each other. The only issue here is that Microsoft Explorer 4.0 (and above) uses a different file format for embedding fonts from the file format used by Netscape 4.0 (and above). Currently, Nuun-encoded web pages embed both font formats. Thus providing browser independent support. Several web sites have been encoded in Nuun and are available for viewing on the browsers mentioned above [www.nuun.net]. Some tools have been implemented to help create documents in and convert across encodings to and from Nuun encoding. One such tool is NuunPad, a Windows program that enables creating and converting Arabic text in CP-1256 to Nuun encoding and vice versa. NuunPad uses the Nuun encoding internally and can run on any version of Windows. Also, we are currently developing, NuunBuilder, a tool that will enable converting already existing Arabic HTML pages to Nuun-encoded Arabic HTML automatically.

Beside the obvious advantages of Nuun encoding platform and browser independence, Nuun provides a very efficient way to view Arabic documents since documents will be read as text not images. We compared file sizes of some of the documents under Nuun’s web page with their sizes if they were presented as simple images (GIF format with two colors: black and white) and as anti-aliased images (GIF format with 15 colors: black, white and shades of gray). Note that anti-aliasing Nuun’s font is free since it is provided by the browser. Beside the fact that the images are limited in being static and hard to modify (to add hyperlinks or underlining for example), Nuun-encoded documents are much smaller than the image-encoded documents. One overhead for loading a Nuun document is loading the Nuun font. This is not included below for comparison because the font is downloaded once and is kept in cache for use by other documents. The following table compares the average number of bytes used in encoding the documents in our test.

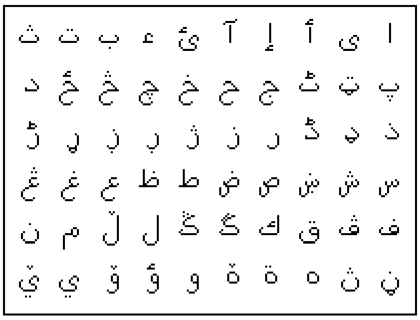
Bare Text	Nuun HTML	GIF	Anti-aliased GIF
3648 bytes	7702 bytes	43,264 bytes	96,704 bytes
1 * Bare Text	2.11 * Bare Text	11.86 * Bare Text	26.5 * Bare Text

Table 1

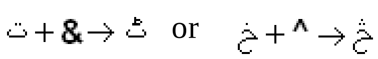
Another advantage of Nuun encoding and fonts is their allowing for early arabization of applications where text editing is a part of the application such as PhotoShop or 3D studio. Users would use NuunPad as a supporting program to create text that can be pasted into these applications. Nuun's web page [www.nuun.net] contains some demos created using Nuun together with 3D Studio and Shockwave. Using NuunPad provides a much simpler solution than having to create application-dependent plug-ins. All Arabic examples in this paper were created using NuunPad on Word 97 under a regular Windows NT.

Of course, there are some limitations to Nuun encoding. For example, the processes of searching and sorting are more complicated in Nuun because of the multiple forms -- (although technically searching is only a level of magnitude more complex than that process in English which if we think of capital and small letters as allographs is an allographic encoding itself.) Another issue is that Nuun encoding is not currently a recognized standard for which main stream applications such as Arabic Word provide support. Because of these limitations, we think of Nuun encoding as a supporting encoding, not a competing replacement, that fills a gap where current encodings fall short.

NuunForm



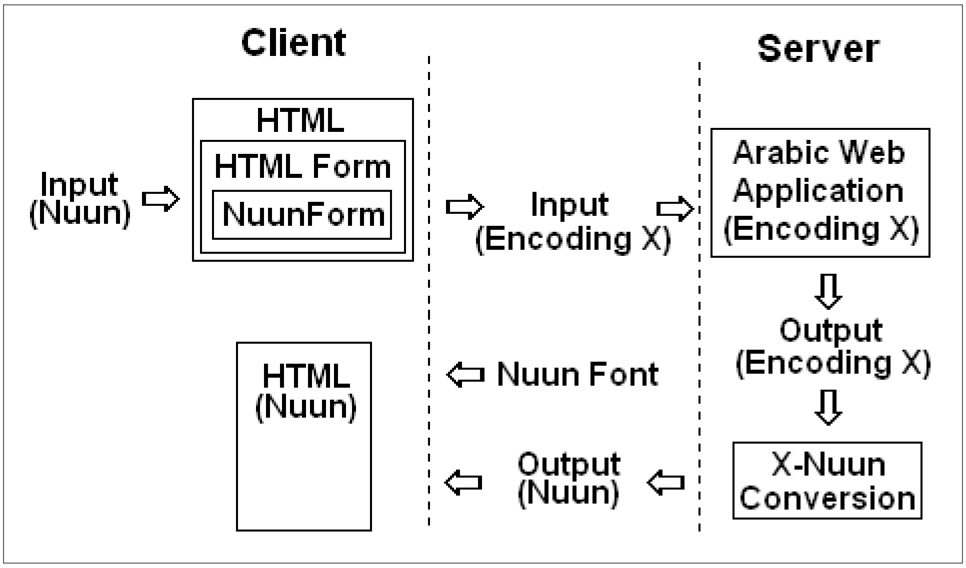
NuunForm is a general purpose input method for Arabic implemented as a Java applet to provide platform and browser independent support for Arabic data entry. It uses a simple bitmap font that doesn't need to be downloaded as it is sent with the applet to the client. NuunForm can be embedded inside HTML forms and it has methods that allows accessing its contents in different encodings. The applet parameters control the size and initial content of the applet in a similar fashion to HTML form text boxes. NuunForm supports two keyboard layouts: Arabic Windows and Nuun Phonetic Keyboard. It also supports input of characters of eight languages extending the Arabic script: Persian, Kurdish, Uigur, Azerbaijani, Pashto, Urdu, Sulu and Malay (see above for a list of currently supported characters). Entering some of the extended characters is allowed through combination sequences. For example,



Multiple instances of NuunForm can be used in the same page without increasing the loading time of the applet. Some examples of NuunForm are shown in Graph 4.

Nuun-Enabled Arabic Web Applications

The different elements of the Nuun toolkit, Nuun encoding and NuunForm, can be used to build platform and browser independent interfaces for Arabic web applications. The term *web application* here encompasses simple HTML documents to highly interactive web accessible databases and applications such as Search Engines, Web Machine Translation systems, eBusiness and Web Mail. Already existing web applications using any standard Arabic encoding can be easily extended to use Nuun on the interface level only without changing any of the application's functionality or databases.



Graph 3

NuunForm allows access to entered data in different Arabic encodings. Accordingly, as far as input is concerned, the web application doesn't see any difference (see Graph 3). Actually, NuunForm can be used without Nuun encoding on systems supporting Arabic display but not data entry such as Windows 98. As for the issue of Arabic display, the Arabic web application will have to convert its output to Nuun's encoding before (or after) formatting the HTML document to be sent as a reply to its input (see Graph 3). Graph 4 shows an example of an interface to an imaginary Arabic Online bookstore.

Future Work

Nuun tools have been tested on small scale applications and have been used in several web sites. The true test of viability is its ability to acquire presence on the World Wide Web by filling the gap in Arabic support. To this end, more work needs to be done to improve the Nuun tools and also a wider variety of font styles needs to be created. Another goal is expanding the Nuun encoding to provide display support to languages extending the Arabic script such as the ones supported by NuunForm.

Graph 4
This is an example of a Nuun-enabled interface for an Arabic Online bookstore. Hypertext, font color and alignment features are controlled through HTML. The HTML form in the lower part of the window contains basic English input boxes and NuunForm text boxes. The alignment of radio buttons to the right of their labels is done using tables. And the Arabic "Send" button is an image. Note that this page is read on a non Arabic supporting platform and browser.

References

Hall, Marty. *Core Web Programming*. Prentice Hall PTR: Upper Saddle River, 1998.

Maeda, Akira, Takehisa Fujita, Lee Swee Choo, Tetsuo Sakaguchi, Shigeo Sugimoto, and Koichi Tabata. A Multilingual Browser for WWW without Preloaded Fonts. University of Library and Information Science, Japan.

Naim, Mohammed C. "Arabic Orthography and Some Non-Semitic Languages." Islam and its Cultural Divergence. Ed. Girdhari L. Tikku. University of Illinois Press: Chicago, 1971.

The Unicode Consortium. The Unicode Standard: Worldwide Character Encoding. Addison-Wesley: Reading, 1992.