GT2004Torsten

Supporting non-Latin character sets with Cocoon

Torsten Schlabach

Interim CEO of PAIWASTOON Networking Services Ltd. Based in Kabul! He works from Europe though.

Paiwastoon is a privately held startup company. Internet is taking off quickly. Prices are still high, but the young generation is very keen on using it.

Non-latin scripts

- In the european union characters a-z will pretty much do. Basic concept of writing is the same.
- Other languages use entirely different character sets, greek, cyrillic. In that case the letter look different but the concept of putting letters next to eachother is still the same.
- Next level of complexity is reached by languages that have so many different characters that they don't fit in 8bit anymore, like chinese, japanese, ethiopic. Even 'simplified' Chinese has about 8,000 characters.
- Next level of complexity is different directions of writing. Hebrew for example is written from lower right to upper left
- One more issue. Shape of characters changes depending on character next to it.

Good example: BBC world service http://bbc.co.uk/worldservice in 43 different languages. (shows demo in MS Word supporting pashtu language in the word 'Salam')

Why Cocoon

- Java and XML both support Unicode well. XML implementations might be an issue.
- Other web languages:

** Perl: do what you like, but you're on your own
** PHP: it tries hard, but falls short in a number of areas.

Content delivery

Means everything on your server from request to response that you control. You don't control the client, the users browser. It is more important than for instance the servers OS. Fonts that user has installed or not on the system.

Treat all encodings except UTF-8 to be deprecated. Basically a compressed version of real Unicode. References on the last slide in the handout.

Shows view->Page Info in Mozilla, which can be used to confirm encoding of the page you're viewing. Demos mistake on aljazeera website, encoding is UTF-8 but content-type charset is specified as windows-1256 in meta-tag. Wrong! BBC Worldwide gets it right. If the encoding is wrong you can use your browser to try and fix it, but you should make sure that meta-tags and http-headers don't contradict. Make sure you control the HTTP headers. Use a 'if you cannot read this page ' button, which links to hints page.

Content Editing

Not really a Cocoon topic. Comparison of opening page source in different editors. JEdit surprisingly doesn't support UTF-8 very well, even though it's a Java app. A good editor for non-latin languages is SimRedo, also on last page of slides.

Shows another page with pashtu language, which also uses numeric entities. Another editor called babelpad is good for converting numeric entities to characters.

Most font problems can only be solved by providing a font to download for the user. Automatically downloading fonts is hard to support in a cross browser way