A GUIDE TO USING PICTOGRAMS IN MOBILE WEB APPLICATIONS

Eduardo Casais
areppim AG
Bern, Switzerland

1. A LONG-STANDING FEATURE

Developers inspecting the user agent profile of a modern handset like the Motorola XT682 ATRIX TV may be surprised to discover the following declaration:

```
<prf:ImageCapable>Yes</prf:ImageCapable>
```

This clause is a remnant from a time when mobile phones boasted displays that showed just a few lines of monospaced characters and could not render bitmap images. As a remedial feature, mobile clients included a set of glyphs conveying stereotypical representations of actions, states and objects. These graphical symbols, frequently stored as a special font in the terminal, could be referred to in Web pages and served as a substitute for richer, but unprocessable pictures. HDML, a markup format for the mobile Web elaborated in the mid-1990s, offered a syntactic construct to this effect (http://www.w3.org/TR/hdml20-6.html#HEADING6-170).

Pictograms were soon considered to be useful as such, and made their way into i-Mode, WAP and UNICODE specifications.

Pictograms differ from icon resources of native smartphone applications:

- They are not specific to a single program, but are standard symbols accessible from all Web applications. The drawback is that their pre-defined style does not allow for a brand or organization-specific look-and-feel.
- They do not have to be explicitly installed, for they are built in the client software. Relying upon pictograms potentially reduces network traffic, as the corresponding images need not be downloaded to the terminal.
- Because they are usually handled like characters, it is straightforward to embed them in a normal text flow. On the other hand, they are often unsuitable as components of a graphical user interface.

Whether in household appliances, packaging, or car dashboards, pictograms are ubiquitous in daily life; their application in mobile services is thus a natural step.

2. APPLICATION

Symbols with easily recognizable semantics can improve the usability of mobile Web applications when used judiciously, for example in the following cases:

1. Distinguishing similar elements from each other.

In the mobile Web, a clickable link may serve to navigate to another Web page; to initiate a file download; to establish a telephone call or a video call; to send an SMS, an MMS or an e-mail. A pictogram identifies the type of action corresponding to each URL.

2. Drawing attention to specific items.

   A pictogram makes elements with important properties stand out:

   - marking items in a list which have been recently modified;
   - tagging those form fields that contain erroneous values;
   - signalling that the activation of a button or a link entails a side-effect.

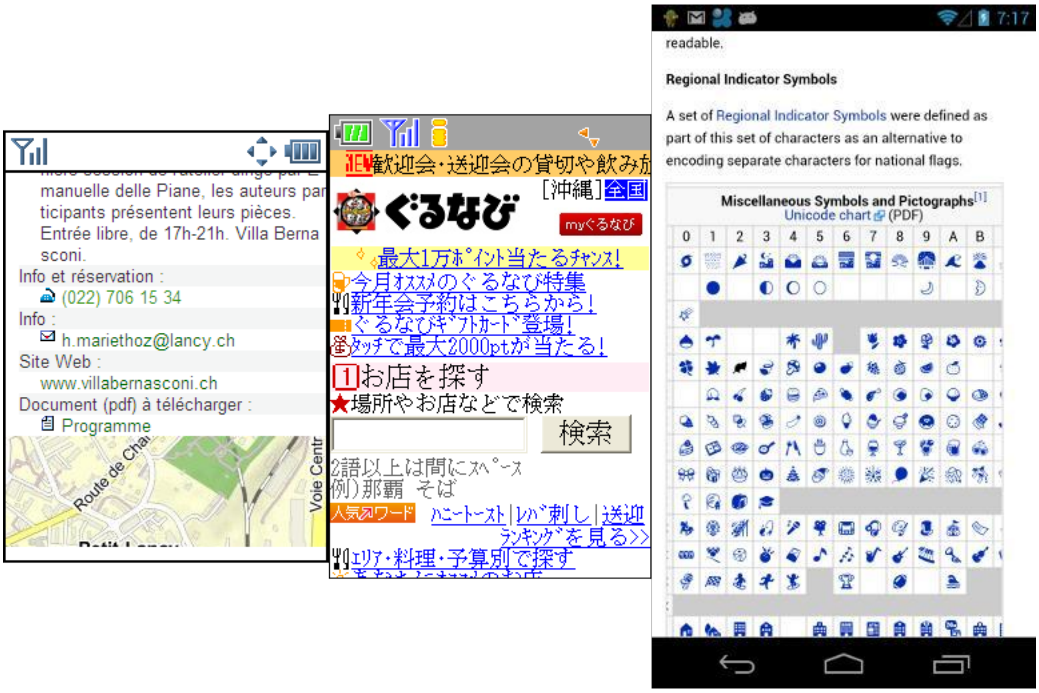
3. Summarizing information.

   If a picture is worth a thousand words, a pictogram is possibly worth a short sentence – as demonstrated by entrenched practices and conventions:

   - railway timetables rely upon symbols to condense instructions such as “seat reservation is compulsory”, “runs only on Saturdays”, etc;
   - tourist guides summarize the description of an accommodation with symbols for the hotel category and its amenities;
   - meteorological forecasts provide overviews constructed out of images for each weather condition (e.g. “heavy snow shower during the day”).

4. Expressing concepts.

   Pictograms act as signifiers so that textual expressions of those concepts are never employed. Well-known instances are emoticons and traffic signs.



manuelle delle Piane, les auteurs par
ticipants présentent leurs pièces.
Entrée libre, de 17h-21h. Villa Berna
sconi.
Info et réservation :
☎ (022) 706 15 34
Info :
✉ h.mariethoz@lancy.ch
Site Web :
www.villabernasconi.ch
Document (pdf) à télécharger :
Programme

NEW歓迎会・送迎会の貸切や飲み放
ぐるなび [沖縄]全国
myぐるなび
最大1万ﾎﾟｲﾝﾄ当たるﾁｬﾝｽ!
今月ｵｽｽﾒのぐるなび特集
新年会予約はこちらから!
ぐるなびｷﾞﾌﾄｶｰﾄﾞ登場!
ﾀｯﾁで最大2000ptが当たる!
1お店を探す
★場所やお店などで検索
検索
2語以上は間にｽﾍﾟｰｽ
例)那覇 そば
人気↗ワード ﾊﾆｰﾄｰｽﾄ|ﾚﾊﾞ刺し|送迎
ﾗﾝｷﾝｸﾞを見る>>
ｴﾘｱ・料理・予算別で探す

readable.

Regional Indicator Symbols

A set of Regional Indicator Symbols were defined as part of this set of characters as an alternative to encoding separate characters for national flags.

Miscellaneous Symbols and Pictographs[1]
Unicode chart (PDF)

The preceding picture illustrates various forms of pictograms. From left to right: a WAP XHTML document with symbols identifying link types; a typical i-Mode page generously spiced with emojis; and a Wikipedia article showing support for ISO pictographs on an Android handset.

Just like with many other aspects of the mobile Web, the deployment of this venerable, albeit little-known building block must cater for various generations of competing standards and discrepancies between implementations. In the following, we survey the ways to declare pictograms in mobile Web pages and provide correspondence tables between relevant symbol dictionaries.

3. OPENWAVE ICONS

The browser developed by the firm Openwave (formerly called Unwired Planet, then phone.com, today Myriad) is the longest serving mobile Web client in existence, and is still deployed on numerous handsets with limited hardware capabilities. It supports a form of pictograms identified by name or by number:

```
<img localsrc=”star2” src=”” alt=”” />
<img localsrc=”69” src=”http://www.site.ch/star.gif” alt=”*” />
```

The syntax works in XHTML and WML. Attribute `localsrc` takes precedence over `src` and `alt`, in both cases displaying the pre-defined icon *star2*. The numbering scheme is recommended. Recognizing an Openwave browser is easy, since its user agent identifier matches the egrep/Perl pattern:

```
(UP\.Browser\/[0-9\.]+)|([Oo]pen[Ww]ave[\/ ]?[0-9\.]+)
```

Version	Icon range	Remarks
≥ 4.1	1 – 175	The browser implements WML
≥ 5.0	1 – 175, 500 – 518	The browser also implements XHTML
≥ 6.1	1 – 175, 500 – 536	
≥ 6.2	1 – 175, 500 – 561	The browser supports WAP pictograms

Nowadays, the original Openwave browser documentation can be retrieved at: http://web.archive.org/web/20071005030630/http://developer.openwave.com/dvl/support/documentation/guides_and_references/.

To this day, the Japanese operator KDDI commercializes phones featuring Openwave software which, depending on the device class, incorporates built-in icon sets supplementing the basic dictionary, or retrieves them as external resources.

Class	Icon range	Remarks
A, B	1 – 175	176 – 330, 500 – 518 as downloadable images; old models
C (1)	1 – 304	305 – 330, 500 – 518 as downloadable images; old models
C (2)	1 – 334, 500 – 518	

D (1)	1 – 334	some models with animated icons
D (2)	1 – 518, 700 – 822	some models with animated icons
D (3), F	1 – 518, 700 – 828	some models with animated icons in D; no animations in F

KDDI publishes on its WWW site a correspondence table between handset models and supported icon ranges. Although all pictograms are assigned names, they are usually embedded in Web pages with the Openwave numbering scheme.

Icons in overlapping ranges are compatible across all Openwave devices, but slight variations exist: several arrows pointing straight upwards or downwards (numbered 42, 43, 70, 71) become oblique in later implementations, whereas some simple triangles become doubly tipped ones (7, 8, 34, 35).

4. WAP PICTOGRAMS

The WAP standard from the Open Mobile Alliance comprises a dictionary of 62 core and 205 optional pictograms, markup to declare them, and a fall-back mechanism in case a symbol is not implemented in the handset. In XHTML mobile profile:

```
<object data=”pict:///core/info/new”>
  <img src=”http://mobile.site.ch/new.png” alt=”NEW” />
</object>
```

The browser outputs the pictogram for *new* if locally available; otherwise, it downloads the file indicated in the `img` tag, or displays the string in the `alt` attribute if the image itself cannot be rendered. Older browsers (such as Motorola MIB) may subject pictograms to the same restrictions as external bitmaps – for instance enforcing a limit on the number of images that can appear on a Web page.

The local presence of specific symbols may vary from a browser version to another; hence, the following table considers whether the client software recognizes the WAP syntax and defaults correctly to an external image when appropriate.

OS / Browser	Vendor	Version	Support	Remarks
Android	Google	≤ 4.1	✗	displays error placeholder
Bada / Dolfin	Samsung	≥ 2.0	⚠	uses the fall-back
Blackberry	RIM	≤ 7.1	⚠	uses the fall-back
iOS / Safari	Apple	≤ 6.0	✗	error alert, empty placeholder
Meego	Nokia	≥ 8.5	⚠	uses the fall-back
MIB	Motorola	≥ 2.2	✓	core set; emotion set ≥ 2.2.1
Mobile IE	Microsoft	≤ 9.0	⚠	uses the fall-back
NetFront	Access	≥ 3.2	✓	implements the core set

Obigo	Teleca	≥ Q05A	✓	implements the core set
Opera Mini	Opera	≥ 4.0	⚠	uses the fall-back
Opera Mobile	Opera	≥ 10.0	⚠	uses the fall-back
S40	Nokia	≤ dp 2.0	✗	displays error placeholder
SEMC	SonyEricsson	≥ 4.0	✓	implements the core set
Symbian	Nokia	< 7.0 ≥ 7.0	⚠ ✗	uses the fall-back displays empty placeholder
TSS	Samsung	≥ 2.5	⚠	uses the fall-back
UP.Browser	Openwave	≥ 6.2	✓	implements the core set

It appears that, while most browsers recognize the syntax declaring WAP pictograms, this feature is fully supported only by legacy browsers not based on Webkit and explicitly preferring XHTML mobile profile over HTML and XHTML basic. This presumably excludes the newest generation of the NetFront (NX), Obigo (v. 10) and Myriad (Openwave v. 9) software built upon Webkit, which could not be tested at the time of this writing. One should also keep in mind that interactive mobile Web validators such as mobileOK (http://validator.w3.org/mobile) and mobiReady (http://ready.mobi) are geared towards XHTML basic, and therefore may erroneously flag WAP pictograms as mobile-unfriendly resources.

The official reference for WAP pictograms is published on the OMA WWW site: http://www.openmobilealliance.org/Technical/release_program/docs/Browsing/V2_3-20080331-A/OMA-WAP-TS-Pictogram-V1_1-20061020-A.pdf.

5. ISO PICTOGRAPHS

ISO normalized numerous pictograms with UNICODE 6.0, refining proposals originally submitted by Google and Apple to harmonize Japanese emojis (see http://unicode.org/charts). The corresponding code points are reserved in various blocks of the UNICODE space, chiefly “miscellaneous symbols and pictographs”, “emoticons”, and “enclosed alphanumerics supplement”; blocks “transport and maps”, “miscellaneous symbols and arrows”, “miscellaneous symbols”, “dingbats”, “enclosed alphanumerics” also contain useful pictograms.

Web pages access all these symbols via numeric character references:

```
&#127775;  <!-- code point U+1F31F represents a star -->
```

ISO pictographs are thus dealt with exactly like characters; there is no graceful fall-back if the necessary font is not installed in the terminal – missing glyphs might appear as question marks, black squares, or blank spaces. Absent a Javascript routine to test the existence of specific glyphs on a client, support for ISO pictograms must be determined on the application server via a device description repository.

Since they are resources of the operating system, pictographs are in principle not tied to the browser and are available to other software modules as well. However, because of their intrinsic nature as characters, some caution is required:

- Several character blocks of interest are located in the UNICODE Supplementary Multilingual Plane. Third-party applications do not always handle characters outside the Basic Multilingual Plane properly. As a consequence, some symbols may be correctly displayed, while others are unrecognised. This appears to be the case with Opera Mobile, which does not render all standard ISO pictographs enabled on Android 4.1.
- Pictographs are usually provisioned through the default system typeface built in the terminal. They may become inaccessible if the user switches his browser or general user interface preferences to a different font.
- Content styled with specific typefaces raises similar problems. Regrettably, CSS offers no construct to force a textual element or a numeric character reference to be rendered in the context of the default client font.

In the following, we consider the pictograms listed in tables of sections 7 and 8. IPR symbols refer to signs for “copyright”, “trademark” and “registered trademark”.

OS	Version	core	emotions	Remarks
Android	≥ 4.1	⚠	✓	Extensive coverage of the ISO norm; a couple of omissions in the core dictionary.
Bada	≥ 2.0	✗	✗	Supports circled digits, IPR symbols and a couple of isolated pictographs.
Blackberry	≤ 7.1	✗	✗	Implements circled digits one to ten, IPR symbols and the single traditional smiley.
iOS	≥ 6.0	✓	✓	Exhaustive provision of “emoticons” and “miscellaneous symbols and pictographs”.
Meego	≥ 1.2	✗	✗	Circled digits, most arrows, IPR symbols and a handful of other pictographs.
Nokia S40	≤ dp2.0	✗	✗	Only implements the IPR symbols and the single traditional smiley.
Symbian	≥ Belle	✗	✗	Implements circled digits one to ten, IPR symbols, and the single traditional smiley.
Windows Phone	≥ 7.5	⚠	✓	Good coverage, except for some blocks like “miscellaneous symbols and arrows”.

With Bada, BlackberryOS, Meego, Symbian and WebOS being either wound down, neglected or undergoing an uncertain overhaul, support for ISO pictographs is likely to remain the preserve of Windows Phone, iOS and Android in the foreseeable future. Only Nokia S40 might be in a position to adopt this feature in the short term.

6. EMOJIS

Mobile applications developed in the Far East rely extensively on pictograms. The Japanese mobile Web is a world unto itself, and its implementations of pictograms (called emojis) exhibit a number of peculiarities:

- Each mobile network operator (Willcom, Softbank, NTT DoCoMo, KDDI) defines its own set of emojis. While dictionaries overlap, there is no bijective relation between them.
- Just like with ISO pictographs, Japanese emojis correspond to symbols in the code space of the document character set. However, because of legacy character encodings, character references vary depending on whether Web pages are formatted in a UNICODE-compliant encoding (such as UTF-8) or not. For instance, the *new* symbol in i-Mode is represented by decimal character reference `&#63874;` in pages encoded with Shift_JIS (corresponding to code point 0xF982 in the Shift_JIS space), but by the hexadecimal reference `&#xE6DD;` in pages encoded with UTF-8.
- In some environments, a pictogram may be directly entered as a sequence of bytes corresponding to its character encoding.
- The set of emojis and their encoding may differ amongst user agents in the same device (for instance the electronic mail client and the browser).

The article at https://technotes.areppim.com/aitceiitmw8/aitceiitmw8.html clarifies the underlying concepts and the intricacies of character encoding.

Operator	Dictionary	Emoji	Representation of symbol "telephone call"
DoCoMo	2 levels 252 symbols max.	📲	`&#63858;` (Shift_JIS) `&#xE6CE;` (UNICODE)
http://www.nttdocomo.co.jp/service/developer/make/content/pictograph/index.html			
KDDI	7 levels 647 symbols max.	📲	`&#xF7DF;` (Shift_JIS) `&#xEB08;` (UNICODE)
http://www.au.kddi.com/ezfactory/tec/spec/3.html			
Softbank	6 levels 485 symbols max.	📲	`chr(27).'$E$'.chr(15)` (PHP string equivalent of Shift_JIS binary code) `&#xE104;` (UNICODE)
http://creation.mb.softbank.jp/mc/tech/tech_pic/pic_material_1.html			
Willcom	2 sets 163 symbols + 252 i-Mode emojis	📱	`&#xF095;` (Shift_JIS, Willcom-specific) `&#63858;` (Shift_JIS, i-Mode compatible)
http://www.willcom-inc.com/ja/service/contents_service/create/emoji/index.html			

Operators have published (mainly in Japanese) detailed documentation about their versions of emojis; they can be obtained from the WWW sites listed above.

Comprehensive correspondence tables describing how to insert emojis into Web pages are presented at http://trialgoods.com/emoji. In 2012, Japanese operators harmonized the design of their pictograms in the style of i-Mode emojis.

Emojis are so popular that Apple was compelled to implement a version derived from the Softbank dictionary upon launching the iPhone in the Japanese market. Version 5.0 of iOS also made these pictograms generally available to non-Japanese iPhone models after a reorganisation to comply with the ISO norm – which implies that pre- and post-iOS 5.0 devices refer to emojis through separate numeric ranges.

iOS version	emojis	Remarks
≥ 2.2.1	469	pictograms stored in a private code space e.g. symbol glowing star: `&#xE335;`
≥ 5.0	468	most pictograms re-mapped to standard UNICODE ranges e.g. symbol glowing star: `&#x1F31F;`
≥ 6.0	845	

The site http://www.iemoji.com lists each iOS emoji with its numeric encodings.

7. CORE PICTOGRAM DICTIONARY

While there is ample on-line documentation about finding equivalent symbols in Japanese emoji dictionaries, such information is sorely lacking for the formats used in the rest of the world. We present here a correspondence table between WAP, Openwave, ISO, and iOS pictograms. The entries are organized according to the core WAP dictionary, and therefore deal with a limited subset of the standards under consideration. This is justified on the following grounds:

- The entire set of potentially relevant symbols defined by ISO inside various UNICODE character blocks amounts to well over 1300 items, and would at any rate result in an unwieldy table.
- Feature phones, most of them running a WAP or Openwave browser, currently represent about 75% of the installed base of handsets worldwide.
- The core dictionary covers the most commonly useful symbols, and, to our knowledge, is the only one to have been fully ported to WAP terminals.

Hence, these tables should prove especially useful when setting up universally accessible mobile Web sites – despite the fact that several pictograms (especially “beginner”, “clear” and “toll-free”) betray their Japanese origins. Developers exclusively targeting modern smartphones need only refer to the ISO norm when inserting pictograms in their applications.

WAP	Openwave		ISO		WP	android	IOS	
Class – name	Nr	Name		Code	7,5	4,1	≥ 5.0	< 5.0
core/arrow								

WAP	Openwave		ISO		WP	android	IOS	
Class – name	Nr	Name		Code	7,5	4,1	≥ 5.0	< 5.0
up	42	uparrow2	⬆	2B06	[7] ✗	✓	✓	E232
down	43	downarrow2	⬇	2B07	[7] ✗	✓	✓	E233
right	70	rightarrow2	➡	27A1	✓	✓	✓	E234
left	71	leftarrow2	⬅	2B05	[7] ✗	✓	✓	E235
upperRight	537	upperRight	⬈	2B08	[7] ✗	2197	2197	E236
upperLeft	538	upperLeft	⬉	2B09	[7] ✗	2196	2196	E237
lowerRight	539	lowerRight	⬊	2B0A	[7] ✗	2198	2198	E238
lowerLeft	540	lowerLeft	⬋	2B0B	[7] ✗	2199	2199	E239
fingerUp	541	fingerUp	👆	1F446	✓	✓	✓	E22E
fingerDown	542	fingerDown	👇	1F447	✓	✓	✓	E22F
fingerRight	141	righthand	👉	1F449	✓	✓	✓	E231
fingerLeft	140	lefthand	👈	1F448	✓	✓	✓	E230
core/button								
1	527	one	①	[2] 2460	✓	[5] ✓	0031 20E3	E21C
2	528	two	②	[2] 2461	✓	[5] ✓	0032 20E3	E21D
3	529	three	③	[2] 2462	✓	[5] ✓	0033 20E3	E21E
4	530	four	④	[2] 2463	✓	[5] ✓	0034 20E3	E21F
5	531	five	⑤	[2] 2464	✓	[5] ✓	0035 20E3	E220
6	532	six	⑥	[2] 2465	✓	[5] ✓	0036 20E3	E221
7	533	seven	⑦	[2] 2466	✓	[5] ✓	0037 20E3	E222
8	534	eight	⑧	[2] 2467	✓	[5] ✓	0038 20E3	E223

WAP	Openwave		ISO		WP	android	IOS	
Class – name	Nr	Name		Code	7,5	4,1	≥ 5.0	< 5.0
9	535	nine	⑨	[2]**2468**	✓	[5] ✓	0039 20E3	E224
0	536	zero	⓪	[2]**24EA**	✓	✗	0030 20E3	E225
core/action								
makePhoneCall	155	phone2	📲	1F4F2	✓	✓	✓	E104
find	119	magnifyglass	🔍	1F50D	✓	✓	✓	E114
userAuthentication	517	personal	🆔	1F194	✓	✓	✓	E229
password	543	password	🔑	1F511	✓	✓	✓	E03F
nextItem	544	nextItem	🔗	1F517	✓	✓	[6] ✓	✗
clear	545	clear	🆑	1F191	✓	✓	[6] ✓	✗
stop	98	stopsign	⛔	26D4	✓	✓	[6] ✓	[1]**E337**
top	35	uptri2	▲	[3]**25B2**	✓	✓	1F53C	E24C
next	8	righttri2	▶	[3]**25B6**	✓	✓	✓	E23A
back	511	back	◀	[3]**25C0**	✓	✓	✓	E23B
✗	34	downtri2	▼	25BC	✓	✓	1F53D	✗
core/message								
receive	546	receive	📨	1F4E8	✓	✓	[6] ✓	✗
send	547	send	📩	1F4E9	✓	✓	✓	E103
message	108	envelope1	📧	1F4E7	✓	✓	[6] ✓	✗
document	56	document1	📄	1F4C4	✓	✓	[6] ✓	✗
attachment	143	paperclip	📎	1F4CE	✓	✓	[6] ✓	✗
folder	79	folder1	📁	1F4C1	✓	✓	[6] ✓	✗
inbox	154	inbox	📥	1F4E5	✓	✓	[6] ✓	[1]**E101**
outbox	153	outbox	📤	1F4E4	✓	✓	[6] ✓	[1]**E102**

WAP	Openwave		ISO		WP	android	IOS	
Class – name	Nr	Name		Code	7,5	4,1	≥ 5.0	< 5.0
core/state								
secure	138	lockkey	🔒	1F512	✓	✓	✓	E144
insecure	548	insecure	🔓	1F513	✓	✓	✓	E145
copyright	81	copyright	©	00A9	✓	[5] ✓	✓	E24E
trademark	54	trademark	™	2122	✓	[5] ✓	✓	E537
underConstruction	549	underConstruction	🚧	1F6A7	✓	✓	✓	E137
beginner	550	beginner	🔰	1F530	✓	✓	✓	E209
✗	82	registered	®	00AE	✓	[5] ✓	✓	E24F
core/media								
book	101	book3	📖	1F4D6	✓	✓	✓	E148
video	115	vidtape	📼	1F4FC	✓	✓	✓	E129
cd	551	cd	💿	1F4BF	✓	✓	✓	E126
dvd	552	dvd	📀	1F4C0	✓	✓	✓	E127
game	45	baseball	🎮	1F3AE	✓	✓	[6] ✓	✗
radio	553	radio	📻	1F4FB	✓	✓	✓	E128
tv	554	tv	📺	1F4FA	✓	✓	✓	E12A
✗	171	newspaper	📰	1F4F0	✓	✓	[6] ✓	✗
core/info								
notice	555	notice	⚠	26A0	✓	✓	✓	E252
charged	14	dollarsign	💲	1F4B2	✓	✓	[6] ✓	✗
freeofcharge	557	freeofcharge	🆓	1F193	✓	✓	[6] ✓	✗
new	72	gem	🆕	1F195	✓	✓	✓	E212
position	556	position	🚩	1F6A9	✓	✓	✓	✗
tollfree	558	tollfree	➿	27BF	[7] ✗	✓	✓	E211

WAP	Openwave		ISO		WP	android	IOS	
Class – name	Nr	Name		Code	7,5	4,1	≥ 5.0	< 5.0
sharpdial	559	sharpdial	#	0023 20E3	[7] ✗	✗	✓	E210
reserved	560	reserved	💺	[4]**1F4BA**	✓	✓	✓	E11F
speechinfo	561	speechinfo	🔊	1F50A	✓	✓	✓	E141
✗	87	creditcard	💳	1F4B3	✓	✓	✓	✗

Values are hexadecimal references to code points in the UNICODE space, and images exemplary depictions of the associated symbols. Sometimes, an operating system provisions the pictogram via a different UNICODE character, whose code point is then mentioned explicitly. We suggest approximate substitutes whenever a straightforward translation from one dictionary to another is infeasible:

1. No direct match; entry is a proposed surrogate.
2. iOS implements these pictograms through UNICODE combining character sequences (numeral followed by an enclosing keycap) – a feature rarely present elsewhere. Relying upon “circled digits” instead is more portable.
3. These symbols are more generic than ISO labelled arrows at code points U+1F51D (“top”), U+1F51C (“soon”), U+1F519 (“back”).
4. ISO character U+1F22F embodies exactly the meaning of “reserved” (e.g. seat reservation), but is a Japanese ideogram. The proposed replacement is more suitable for other countries.
5. Already available in Android version 1.6.
6. Only available in iOS since version 6.0.
7. Consistent implementation only in Windows Phone 8.0.

8. EMOTICON DICTIONARY

Class “emotions” is the sole subset of the optional WAP dictionary that has enjoyed some (modest) level of support in mobile phones. It comprises a number of smileys for which we propose equivalent iOS, ISO and Openwave representations. This table also applies the tweaks and descriptive conventions used for the core dictionary.

WAP	Openwave		ISO		WP	android	IOS	
Class – name	Nr	Name		Code	7,5	4,1	≥ 5.0	< 5.0
emotions								
smile	520	:)	😃	1F603	✓	✓	✓	E057

WAP	Openwave		ISO		WP	android	IOS	
Class – name	Nr	Name		Code	7,5	4,1	≥ 5.0	< 5.0
cry		✗	😢	1F622	✓	✓	✓	E413
sad	521	:(	😔	1F614	✓	✓	✓	E403
angry		✗	😠	1F620	✓	✓	✓	E059
pullFace	525	:P	😜	1F61C	✓	✓	✓	E105
inLove	51	heart	😍	1F60D	✓	✓	✓	E106
shock		✗	😱	1F631	✓	✓	✓	E107
coldSweat		✗	😓	1F613	✓	✓	✓	E108
shakenHeart		✗	💓	1F493	✓	✓	✓	E327
brokenHeart		✗	💔	1F494	✓	✓	✓	E023
discourage	524	:|	😞	1F61E	✓	✓	✓	E058
flash	77	lightbulb	💡	1F4A1	✓	✓	✓	E10F
sleepy		✗	😪	1F62A	✓	✓	✓	E408
anxious		✗	😨	1F628	✓	✓	✓	E40B
surprised	526	:o	😲	1F632	✓	✓	✓	E410
tutting		✗	😒	1F612	✓	✓	✓	E40E
happy	523	:D	😄	1F604	✓	✓	✓	E415
punch		✗	👊	1F44A	✓	✓	✓	E00D
wink	522	;)	😉	1F609	✓	✓	✓	E405
thumbsUp	**26**	[1]**plus**	👍	1F44D	✓	✓	✓	E00E
thumbsDown	**27**	[1]**minus**	👎	1F44E	✓	✓	✓	E421
kiss		✗	😘	1F618	✓	✓	✓	E418
smell		✗	😏	1F60F	✓	✓	✓	E402
cool		✗	😎	1F60E	✓	✓	✓	✗
hug		✗	😚	1F61A	✓	✓	✓	E417
trapped		✗	😳	1F633	✓	✓	✓	E40D

WAP	Openwave		ISO		WP	android	IOS	
Class – name	Nr	Name		Code	7,5	4,1	≥ 5.0	< 5.0
shine		❌	😊	1F60A	✓	✓	✓	E056
❌	68	smileyface	☺	263A	✓	✓	✓	E414

Smileys have long been extremely popular in messaging applications and Internet discussion forums. Every major platform (Yahoo! Messenger, Microsoft Messenger, America Online, among others) offers a rich set of icons to let users liven up the texts they send to each other. Emoticons are entered as pre-defined identifiers or character sequences (for instance :-) or 8-|). The messaging clients of some mobile operating systems (notably Windows Phone) follow a similar approach.

9. SERVICE PLATFORMS

Unfortunately, major device description repositories provide very few data describing pictographic terminal capabilities. Developers may resort to attributes such as the type of browser or the version of the operating system run by the handset in order to fine-tune the representation of mobile pages.

DDR	Attribute	Values	Description
51 degrees	no specific attributes available		
	http://51degrees.mobi/Products/DeviceData/PropertyDictionary.aspx		
Detect Right	supportedpictogramset	array of literals	List of WAP pictogram classes implemented (from UAProf).
	http://www.detectright.com/detectright-data.html		
Device Atlas	no specific attributes available		
	https://deviceatlas.com/resources/available-properties		
Handset Detection	no specific attributes available		
	http://www.handsetdetection.com/resources/apiv3_schema		
WURFL	emoji	true / false	Indicates support for i-Mode emojis.
	opwv_xhtml_extensions_support	true / false	Indicate support for Openwave icons and other proprietary extensions.
	opwv_wml_extensions_support	true / false	
	http://www.scientiamobile.com/wurflCapability/tree		

Programs that map pictogram dictionaries and encodings to each other are necessary when generating content for terminals implementing different pictographic formats, or when exchanging messages amongst them. Here is a sample of public-domain tools that translate between popular Japanese conventions.

Language	DoCoMo	KDDI	Softbank	Willcom	iOS < 5.0	ISO
RUBY	⚠	✓	✓	✗	✓	✗
	Only UNICODE encoding. DoCoMo codes erroneously assumed identical to those of KDDI.					
	http://bitcontrol.org/2009/10/18/emoji-rubygem-for-iphone-softbank-ntt-docomo-kddi					
PHP	✓	✓	✓	✗	⚠	✓
	Only UNICODE encoding. Although not explicitly supported, iOS pre-5.0 uses Softbank codes.					
	http://code.iamcal.com/php/emoji					
PERL	✓	✓	✓	✗	⚠	✗
	Shift_JIS and UTF-8 encodings. Although not explicitly supported, iOS pre-5.0 uses Softbank codes.					
	http://search.cpan.org/dist/Encode-JP-Emoji/lib/Encode/JP/Emoji.pm					

We are not aware of any similar gateway utilities for ISO, Openwave and WAP.

10. CONCLUDING REMARKS

The interest in pictograms was rekindled when Apple released an iPhone model with an abundant collection of colourful icons embeddable in applications and Web pages. The revision of UNICODE published in October 2010 increases the relevance of pictograms for smartphones, but feature phones and legacy handsets have long been endowed with a comparable capability – albeit via different mechanisms. The information presented in this article should enable developers to decide how to integrate pictograms in their Web sites – taking into account requirements about application look-and-feel, the range of targeted devices, and the facilities to configure their Web service platform with pictogram generators.

We recommend a utilization of pictograms more parsimonious than what one can often observe in Asian mobile Web sites. Hence, the core dictionary should suffice for the first couple of – very common – scenarios listed in section 2.

Beyond this basic set, ISO specifies a profusion of idiosyncratic signs for the Chinese horoscope, Japanese places and holidays, Nippon cuisine, service ideograms, manga characters, as well as more universally interesting pictographs related to office supplies, commercial and professional activities, sports, games, vehicles, plants, animals, the weather, geometric shapes and additional emoticons. WAP defines many optional items that map easily to those ISO code points, but lacking support in WAP handsets makes this compatibility immaterial in practice. On the other hand, about 160 actual Openwave icons correspond to symbols scattered throughout UNICODE and may prove occasionally useful when catering for low-end devices. Developers can peruse the Openwave documentation to ascertain the equivalence of individual pictograms on a case-by-case basis. Sites such as http://shapecatcher.com and http://www.fileformat.info/info/unicode/index.htm have query tools that facilitate the exploration of the UNICODE character space.

Of course, the situation may arise where the symbol of interest is completely unsupported by a handset or a standard. The manipulation of custom icons (such as those used in the present article, extracted from the fatcow collection of free icons at http://www.fatcow.com/free-icons) is an issue to be addressed in another instalment.

REFERENCE

Eduardo Casais: *A guide to using pictograms in mobile Web applications*, technical paper, areppim AG, Bern, Switzerland, 2012-10-20, 17 pages.

ABOUT THE AUTHOR

Eduardo Casais has been working on mobile Web technologies since 1997. He led the development of content adaptation facilities in the Nokia WAP Gateway. He was also involved in projects dealing with transcoders for WWW on TV set-top-boxes. Eduardo Casais has been an invited expert to the Mobile Web Best Practices Working Group of the World-Wide-Web Consortium, where he participated in the elaboration of the Content Transformation Guidelines.

ABOUT AREPPIM AG

areppim AG develops Internet applications, with an emphasis on the display of quantitative information. The site http://www.areppim.com publishes data on a wide range of topics, presented as intuitive, content-rich charts and often accompanied by concise analyses. Naturally, data can also be accessed with mobile phones at http://mobile.areppim.com through a no-frills, mobile-optimized interface.

ADDRESS

areppim AG
Wankdorffeldstrasse 102
P.O. Box 261
CH-3000 Bern 22
Switzerland

e-mail: info@areppim.com