

A new transcoding technique for PDA browsers, based on content hierarchy*

F. J. González-Castaño, L. Anido Rifón, E. Costa-Montenegro

Departamento de Ingeniería Telemática, Universidad de Vigo, ETSI
Telecomunicación, Campus, 36200 Vigo, Spain
{javier, lanido, kike, jvales, rasorey}@det.uvigo.es

Abstract. This paper presents a new transcoding technique for WWW navigation on small display devices: Hierarchical Atomic Navigation. Unlike previous techniques, Hierarchical Atomic Navigation keeps all original information in a readable way, without imposing the use of a specific browser. To achieve this goal, a *navigator page* is used to represent original contents in a symbolic way. A set of representative icons replaces unreadable elements. These icons are linked to actual individual contents, as a set of atomic pages. Hierarchical Atomic Navigation can be used on any PDA, regardless of OS and browser choice, since both navigator and atomic pages use widely supported standard formats (e.g. XML, HTML).

1 Introduction

In the future, many WWW services will be accessed from cell phones and wireless Personal Digital Assistants (PDA) [1–4]. Although GPRS and 3G systems may provide enough mobile bandwidth, there is an intrinsic problem to be solved: mobile terminals are small and, therefore, have small displays. Most WWW contents are not designed for them [5, 6]. Searching information through a page is a complex task, users are prone to incorrect selections and scroll up and down most of their time. It has been reported [6] that users with small screens are 50% less effective than users with large screens. A trivial (and obviously undesirable) solution is limiting mobile WWW services to specialized WAP-like ones, like stock quotes, weather forecasts and sports scores.

This paper presents a new philosophy to improve WWW navigation on small displays. Our system, described in section 3, is based on the Hierarchical Atomic Navigation concept (HAND): some or all elements embedded in a WWW page are identified by means of a reduced page preview, the *navigator page*, which is generated automatically. Section 2 discusses related work. Section 4 evaluates usability. Finally, section 5 concludes.

* Additional authors: J. Vales-Alonso, R. Asorey-Cacheda, J. García-Reinoso, D. Conde-Lagoa, J. M. Pousada-Carballo and P. S. Rodríguez-Hernández. This research has been supported by European Commission FEDER grant 1FD97-1485-C02-02 TIC.

2 Related work

2.1 Web clipping

Web clipping is based on delivering a “clipped” version of the original WWW page to a small display device. First, a page fragmentation process is carried out. Then, a human or an automated process specifies an importance value for every page fragment. Low importance fragments are ignored when display space is limited. For example, elements that do not carry significant information in relation with their size.

Hori et al. designed a page fragmentation system based on external annotation files [7]. A transcoding process uses annotations to decide which elements will be delivered to the client. Annotation files follow the RDF data model [8] to specify the relative priority of page elements (punctuation). The system is basically composed of a page splitting module and a generator, which creates the client page. The system includes an authoring tool, to create annotation files with page fragment priorities.

Although clipped versions of WWW pages are adequate for small displays, there is information loss according to criteria that are ignored by the end-user. Also, page fragments must be punctuated with their priority (this process may be accelerated by means of advanced authoring tools [7]).

2.2 Handy-Fit-to-Screen

Microsoft Pocket Internet Explorer [9] implements the Handy-Fit-to-Screen feature [10], which resizes WWW pages to fit into small displays. Obviously, many page elements are not properly displayed. To cope with this drawback, Pocket Internet Explorer includes a zoom menu option that displays text in several sizes.

Handy-Fit supports typical commercial standards, including HTML and XML [11], and keeps all information. Although it is a particular feature of Microsoft Pocket Internet Explorer, similar tools could be developed for other browsers. Nevertheless, it does not work properly on large text pages that use small fonts.

2.3 WML

WML WAP devices [12] use microbrowsers to access WAP-supporting WWW sites. WML is designed to optimize text delivery over limited-bandwidth wireless networks. It supports WMLScript, which is similar to EcmaScript, but imposes minimal system resource demands. It is unlikely that WML will handle color, audio and video in the near future.

Typical WAP sites present a series of options for visitor access. Whilst clearly limited, this operation mode is satisfactory for the delivery of news, stock quotes, sports results, travel inquiries and similar applications. However, it cannot support the capabilities of most WWW sites. A WML version of a WWW page is, in a sense, a “clipped” version for the WAP-enabled cell phone world.

2.4 i-MODE

In February 1999, the wireless branch of Nippon Telegraph & Telephone, NTT DoCoMo, launched the i-MODE network [13]. By May 2000, the service had more than seven million subscribers in Japan and the demand was so great that NTT stopped advertising for a while, in order to install extra service capacity. i-MODE terminals access WWW contents over the PDC wireless network, using a proprietary DoCoMo protocol.

A typical i-MODE terminal has a 95×65 -pixel screen, which is sharper than most current cell phone resolutions. In order to display WWW contents on such a small screen, i-MODE employs subsets of HTML 2.0, 3.2 and 4.0. These subsets are called c-HTML [14], which stands for compatible-HTML. This language is the only requirement for porting a WWW site to the i-MODE system. Since c-HTML relies on ordinary HTML coding and uses common HTML tags, c-HTML pages can also be displayed on PCs. i-MODE terminals can display GIF images and half-width Kana characters, but do not support Java or script languages.

Unquestionably, i-MODE is a successful experience in the Japanese market. Its main advantage is the lack of punctuation processing, since clipping is an implicit consequence of using c-HTML. Consequently, an i-MODE WWW site is just a “clipped” version of the original site. All elements outside the subset are no longer included in the i-MODE page and, therefore, no longer viewable by i-MODE terminals. Again, we are facing the problem of information loss, according, in this case, to fixed criteria over a proprietary network.

From this analysis, it can be concluded that most approaches rely on some form of clipping, which yields high readability but has information loss. Microsoft Handy-Fit avoids that disadvantage, but it is not adequate for large text pages. In the next section, we present a new philosophy that keeps all original information in a readable way, and does not impose a browser choice.

3 HANd

3.1 Philosophy

The HANd philosophy is based on the fragmentation of the original page into zones. These zones are sets of multimedia elements included in the original page (e.g. images, text paragraphs, headers, hyperlinks, Java applets, forms, etc.). The user may select any zone, but only one can be displayed at original scale at a given time.

The main page, or navigator page, is a reduced overview of the original page. The navigator page is always displayed on a side frame or pop-up window. It facilitates access to different atomic readable elements (i.e. pieces of the original page), using links to auxiliary or *atomic pages* (on a second side frame or pop-up window, according to user preferences and display capabilities). There are different navigator page representations for a given element in the original WWW page:

- A reduced version of the element, if still readable.
- A representative icon, if it is decided that the reduced element is unreadable.

A key issue is element dependence. All original elements are ordered in a tree hierarchy, and any element is considered unreadable if all its descendants are unreadable. In the navigator page, the representative icon of an ancestor hides all its descendants, but, if that icon is selected, all descendants are shown on the atomic page.

Navigator page representatives are placed in the same relative positions the corresponding elements occupied on the original page. The goal is to provide the user with an *idea* of the original appearance of the page, in a reduced space.

Note that HANd can be considered a sort of visual clipping (see section 2). However, unlike web clipping techniques, every single information item is kept and the end user is free to decide which elements are shown.

Also, HANd can be used on any PDA, regardless of OS and browser choice, since both navigator and atomic pages use standard formats. This is a major difference with Microsoft Handy-Fit, a technique that is implemented by specific browsers.

3.2 HANd page examples

Figures 2 to 5 show the navigator/atomic page set after processing the page in figure 1. All figures are captures taken on an iPAQ H3630 PDA. The width ratio between figures 2 to 5 and figure 1 on a real screen is approximately 20%. The left screen in figures 2 to 5 shows some possible user selections on the navigator page. The right screen shows the corresponding atomic pages (the “atom” icon is used to return to the navigator page, in an implementation without frames).

Figure 6 shows a selection of representative icons used in navigator pages. On a real PDA screen, the icons are blue if the corresponding atomic pages contain hyperlinks.

3.3 HANd page generator prototype

The previous section described a navigator/atomic HANd page set. These pages are created by a HANd page generator, which is shown in figure 7. The generator receives documents written in any language, and processes them in two stages:

- *Preprocessing stage.* Non-XML documents (HTML, TEX,...) are converted into XHTML, which is a reformulation of HTML 4.01 in XML. The conversion is performed by JTidy [15], the Java version of HTML TIDY [16]. Once a XHTML document is available, we obtain its DOM tree [17] using a DOM parser. A DOM tree is a representation of a XHTML document as a tree of nodes, each one corresponding to a element in the XHTML document.
- *Conversion stage.* This stage applies visibility rules to the DOM tree, and obtains the navigator/atomic page set from the resulting new DOM trees.



Fig. 1. Original page (it occupies a full 17" 1024x768 screen)



Fig. 2. Navigator/atomic page set



Fig. 3. Navigator/atomic page set



Fig. 4. Navigator/atomic page set



Fig. 5. Navigator/atomic page set

	List with hyperlinks
	Hyperlinks
	Table row with text and hyperlinks
	Paragraph with hyperlinks

Fig. 6. Representative icons

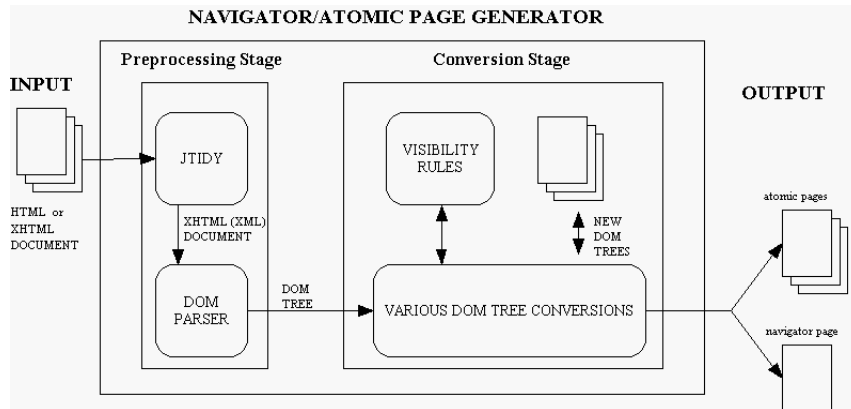


Fig. 7. Navigator/atomic page generator

DOM generates the tree hierarchy required in our philosophy and the new trees to obtain the navigator/pageset. For non-XHTML WWW contents we need a JTIty preprocessing stage to generate the appropriate input for the conversion stage. Note that this conversion does not impose constraints on original WWW contents (this is not the case of WML and c-HTML).

A final remark: HANd implementation is based on language conversion, in a broad sense. It can be considered a particular case of transcoding strategy [7].

3.4 Transformation rules and representatives

The following rules associate original page elements to their representatives in the navigator page. The tree of the original page is followed in increasing hierarchical order, ending at the root node, n_s . Let (n_i, n_{i-1}) denote a parent-child node relationship between levels i and $i-1$, $i = 2, \dots, s$.

For a XML page that has not been explicitly created for a small display device, it is assumed that a generic large display produces a good presentation. Using a 17" 1024×768 screen as a reference, for a specific PDA, the PDA-to-PC screen area ratio $R < 1$ is calculated. This ratio is used to reduce the size of all elements in the original XML page. In case of local processing, the user could set R manually.

Then, the PDA screen area occupied by leaf nodes (without descendants) is calculated. For the sake of clarity, we comment the most typical cases, although it is easy to find similar rules for other elements:

- *Text*: Since we are interested in readability, the font size for the PDA screen (after reduction) is calculated.
- *Images*: Given the PDA screen resolution and the (reduced) image size in pixels, the occupied PDA screen area is estimated.

For any reduced element, a readability threshold is applied. If a reduced element is smaller than its threshold, it is considered unreadable, and is labeled with 1. Otherwise, it is labeled with 0. There are some exceptions: for example, HR lines are always unreadable. Let $l(n_i)$ be the label of node n_i . Then, element labels are propagated across the XML tree as follows:

For $i = 2, \dots, s$, do:

For all n_i , do:

Let I_{n_i} be the set of all nodes n_{i-1} such that (n_i, n_{i-1}) holds. Then, $l(n_i)$ is the product of the labels of all nodes in I_{n_i}

Finally, all branches that do not contain a 0 label are pruned. For the resulting subtree, the following rules apply:

- All nodes whose labels are 0 (readable nodes) are substituted in the navigation page by reduced representatives.
- All nodes whose labels are 1 are replaced by an icon. There is a different icon for each element (tables, lists, text...). For a given element, there are two possibilities, with a different background color: normal icon (black) and hyperlink icon (blue, if the original element is associated to a hyperlink). Specific icons are resized so that their area is approximately equal to the screen area covered by all associated descendants.
- Each icon is linked to an atomic page, containing either the associated original element or *all original elements associated to the corresponding pruned branch*, for intermediate nodes.
- *Adjacency rule*: all adjacent icons associated to elements of the same type are represented by a single icon. Elements such as HR or BR are transparent when applying this rule. Also, other information-irrelevant elements like small pictures (bullets, etc.) are considered transparent.

Figure 8 shows the three stages of the algorithm for a simple tree example: (a) labeling, (b) label propagation and (c) pruning.

Figure 9 shows the hierarchy tree of the page in figure 1. Note that several elements are pruned and replaced by the corresponding representative icon. The adjacency rule applies to the object group in the dotted box.

4 Usability tests

In order to test the effectiveness of the new transcoding technique, we performed usability tests based on the WWW site that starts with the page in figure 1. The site contains a user manual that explains how to use the remote access system to a third-year laboratory course, *Arquitectura de Ordenadores II*.

We divided a group of students into three testing groups. The first one accessed the WWW site from the desktops in the laboratory room. The second

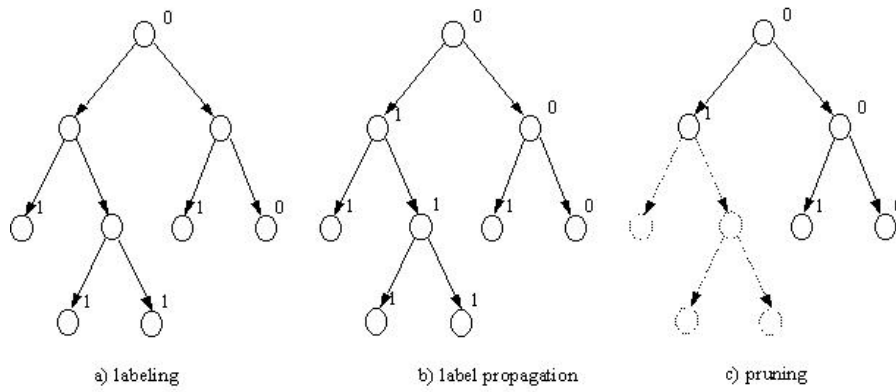


Fig. 8. Basic transformation algorithm

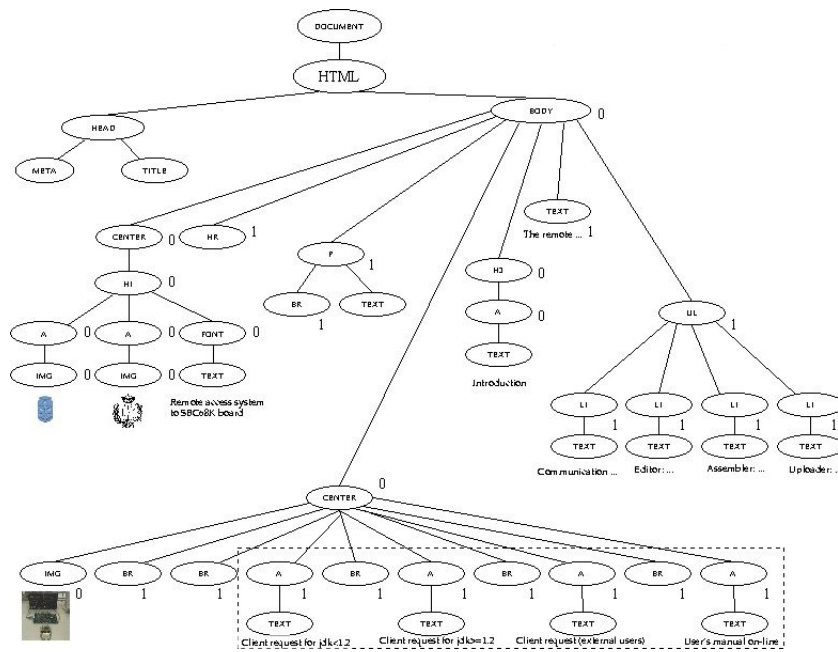


Fig. 9. Hierarchy tree of the page in figure 1

one accessed the WWW site from an iPAQ H3630 PDA. The third one accessed the transcoded version from an iPAQ H3630 PDA. They had to solve a quiz on laboratory usage.

We tracked two quality measures per group: *average success* (percentage of right answers) and *average access time* (time to finish the quiz).

In order to calculate access time, we served all pages (original and transcoded) with Apache. Apache traces follow the CLF standard. Each line displays the IP address of the accessing computer, GMT access date with seconds-resolution, submitted command (GET, POST ...) and server answer (OK, 404 ...). Individual user times can be easily extracted with a filter.

As we expected, it was hard for group 2 to visualize normal pages on a PDA display. This made users prone to commit more mistakes. The first time the users in group 3 accessed the transcoded pages, their access time was slightly longer than the access time in group 1, but success was similar (and clearly better than success in group 2). In subsequent tests, the performance of users in groups 1 and 3 was similar, which suggests that users got used to the transcoding technique easily. Nevertheless, there was no improvement in group 2, neither in success nor in access time.

It should be understood the results cannot be generalized, since the users were skilled students. However, we think that they demonstrate the potential of the approach for small-to-medium traditional pages.

5 Conclusions and future research

In this paper, we have presented a new transcoding technique for small display devices: Hierarchical Atomic Navigation (HANd). Like the alternative transcoding techniques in section 2, it may need manual adjustment when processing visually complex pages like Amazon.com or www.elpais.es. Nevertheless, it has the following advantages over them:

- No information loss. All original elements can be shown on demand, as atomic pages.
- The navigator page is a readable overview of the original page, using a reduced space. Actual contents are accessible via their representatives (atomic navigation).
- Atomic navigation can be adapted to any WWW document, including pages created by accessing ASP or PHP pages. There is no need for a mark-up language that introduces content constraints, or a specialized WWW browser (both navigator and atomic pages are supported by standard mark-up languages).

Current work is oriented towards completing the following objectives:

- Integration of script languages in the framework. A language processor is required to generate a hierarchy of explicit objects. This could be achieved by means of tools like Rhino [18].

- Further enhancements of the HANd philosophy. For example, design of rules to generate hierarchies of navigator pages (or, in other words, a representative icon could be linked to a new navigator page in a lower hierarchical level).

References

1. “Market Risks: Security: The Downside of .com”. Research briefs.
<http://www.infoworld.com/articles/hn/xml/99/07/19/990719hnrbc.com.xml>
2. PDAs at PC technology guide. <http://www.pctechguide.com/25mob3.htm>
3. Personal mobile communicators. <http://www.pdastreet.com/>
4. Lewis, T. “UbiNet: the ubiquitous Internet will be wireless”. IEEE Computer 32(1999), pp. 126-128.
5. Gessler, S. and Kotulla, A. “PDAs as mobile WWW browsers”. Proc. of the 2nd Int. WWW Conference. Chicago, USA. October 1994.
6. Jones, M., Mardesen, G., Mohd-Nasir, N., Boone, K. and Buchanan, G. “Improving Web interaction on small displays”. Proc. of the 8th Int. WWW Conference. Toronto, Canada. May 1999.
7. Hori, M., Kondoh, G., Ono, K., Hirose, S. and Singhai S. “Annotation-based web content transcoding”. Proc. of the 9th Int. WWW Conference. Amsterdam, The Netherlands. May 2000.
8. Resource Description Framework (RDF). <http://www.w3.org/TR/REC-rdf-syntax/>
9. Microsoft Pocket Internet Explorer.
<http://www.microsoft.com/mobile/pocketpc/features/pie.asp>
10. Handy-Fit-to-Screen.
<http://www.microsoft.com/mobile/pocketpc/features/articles/web.asp>
11. XML at W3C site. <http://www.w3.org/>
12. WAP. <http://www.wapforum.com/>
13. NTT DoCoMo. <http://www.nttdocomo.com/>
14. c-HTML. <http://www.nttdocomo.com/source/tag/index.html>
15. JTidy. <http://lempinen.net/sami/jtidy/>
16. HTML TIDY. <http://www.w3.org/People/Raggett/tidy/>
17. DOM at W3C site. <http://www.w3.org/DOM/>
18. Rhino at Mozilla site. <http://www.mozilla.org/rhino/>