



NOTE-Submission-HDML

Proposal for a Handheld Device Markup Language

· [Submitted to W3C](#) on 09 May 1997 ·

This version:

<http://www.w3.org/pub/WWW/TR/NOTE-Submission-HDML.html>

Latest version:

<http://www.w3.org/pub/WWW/TR/NOTE-Submission-HDML.html>

Author:

[Tim Hyland](#), Unwired Planet

Status of this Document

This document is a NOTE made available by the World Wide Web Consortium for discussion only. This indicates no endorsement of its content, nor that the Consortium has, is, or will be allocating any resources to the issues addressed by the NOTE. A list of current NOTES can be found at:

<http://www.w3.org/pub/WWW/TR/>

This document is part of a complete [submission](#) to the W3C. The full submission has been acknowledged by W3C and is available at

<http://www.w3.org/pub/WWW/Submission/1997/5/Overview.html>

Note: since working drafts are subject to frequent change, you are advised to reference the above URL, rather than the URLs for working drafts themselves.

1.0 Introduction

The World Wide Web provides a robust, flexible, and ubiquitous model for information access. The adoption of the WWW as the preferred means of disseminating and accessing information from desktop PCs and workstations has created a demand for access to the same information for other devices. These devices or "alternate platforms" range from voice- and fax-based user agents to low-cost Network Computers to handheld devices such as mobile phones and PDAs.

While the web's infrastructure and protocols fully support these alternate platforms, HTML itself does not. In particular the navigation and display models inherent to HTML collapse when applied to a typical handheld device. However, combining the use of standard web protocols and infrastructure (URLs, HTTP, SSL plus CGI, Perl, commercial web servers) with an alternate but complementary markup language, allows handheld devices to function as full-fledged web clients.

This proposal motivates and proposes a mark-up language specifically for handheld devices, called the Handheld Device Markup Language (HDML). HDML provides display and navigation models designed to provide intuitive data access using a modified transaction paradigm rather than the document-based paradigm of HTML.

2.0 Handheld Devices

While there are many types, styles, and classes of handheld devices, the specification proposed in this document is useful for a significantly larger class of devices with similar physical characteristics. Those characteristics include:

- small display size (always)
- limited input capabilities (often)
- limited bandwidth both (often)
- limited resources such as memory, processing power, permanent storage (often)

This proposal will use the data-ready mobile phone as an example of the typical handheld device. This proposal will also continue to use the term "handheld," recognizing that the term is not inclusive of all devices which would benefit from HDML.

2.1 Physical characteristics

Handheld devices are characterized primarily by a limited display size. A typical display is capable of displaying 4-10 lines of text 12-20 characters wide and may be graphical (bitmapped) or text-only. For purposes of this proposal, PDA-style displays are not included in this handheld device category, although the markup language proposed will be useful on those devices as well.

Handheld devices may or may not have a full keyboard and may or may not have a pointing/selection device. As an example, the data-ready mobile phone has only:

- the keys normally found on a telephone (0-9, *, #, with alphabet letters marked on 2-9)
- cursor/arrow keys (often just up and down or left and right)
- a number of dedicated function keys (SEND, END, etc.)
- one or more "soft keys" with programmable labels

Alphanumeric pagers have even fewer keys. Item selection is accomplished through numbered lists or using cursor keys to highlight a choice and then request an action on that item. Full 2-D cursor control through pointing devices such as touch pads, touch screens, roller balls, are rare. A full QWERTY keyboard is also rare. Input is severely constrained when compared to a typical PC or PDA.

Network bandwidth is usually low due to limitations of the network technology or simple economics. The same goes for other resources: memory, processing power, even battery life. All in the name of economics. While some devices do have large amounts of memory or processing power, these devices are the exception. Mass market and consumer-targeted devices will continue to have these constraints

for many years.

2.2 Implications

Because of the smaller displays and limited input, bandwidth, and resources, handheld devices must be viewed as distinctly different from typical PCs, workstations, and other common platforms found on the web today. For a handheld device to access data on the web, that data must be presented in a way which:

- makes sense on a small display
- is easily navigable using the available context and input capabilities
- does not strain the limited bandwidth, memory, or processing power available

Upon examination, it becomes clear that HTML cannot provide a suitable mechanism for accessing and navigating data on handheld devices.

3.0 Limitations of HTML on Handheld Devices

The HTML document model can be considered to consist of structural idioms such as header, paragraphs, lists, tables and figures as well as rendering idioms such as margins, leading, font names and sizes. These structural and rendering idioms are all based on a default window width of about 80 characters. In addition, these structural idioms is often based on the paper-roll layout model in which the user may scroll down forever viewing more and more structural elements. Thus the 80 column wide and paper-roll length virtual display space has been considered the least common denominator for most HTML documents created today. This model has proven to be widely acceptable and portable for any device that can meet these display requirements.

However, one would not want to try to view a typical web page on a handheld device. A number of methods have been proposed (and implemented) to view and navigate web-based data on handheld devices. These methods fall into two categories. The first category attempts to distill an existing page to extract the essential data which can be displayed on a handheld device. This fails for a number of reasons:

- visual data cannot be machine parsed: many HTML pages present data in visual forms not easily reduced or parsed by a machine, image maps being the best example
- loss of data: most attempts at machine distillation will lose data deemed important by the author
- loss of context: the hypertext navigation model of most HTML pages would not translate well to a device with a small display

The second category requires documents to be written using a subset of HTML, using only those components of HTML that are possible to view on a handheld device. While this solves the issue of presenting data on the small display, it does not address the inherent navigation model imposed by HTML. The navigation model in HTML is a simple directed graph. You can link from one document to the next. This model requires a fairly complex and rich user agent to make it useable. Web browsers provide a significant amount of context -- title bars, pull-down history lists, URL displays -- that allow the user to keep track of where he or she is. Navigation becomes a sort of indirect traversal of information relying heavily on visual cues and context.

However, the display constraints of handheld devices do not allow the user agent to provide the same degree of context. (To illustrate this point, shrink your web browser display window to a size where you can see only 4 lines by 12 characters.) In such a context-limited environment, the best way to aid

navigation is to make that navigation predictable, deterministic, and consistent. However, HTML does not provide the tools for building an efficient navigation model which is useful with minimal visual context.

4.0 Basis for a new markup language

Of all the differences and constraints mentioned, two in particular inform the basic design of a new markup language: restricted display size of handheld devices and lack of context for traditional navigation. These parameters indicate an application model in which data is presented in discrete packages which are grouped together to form a coherent set of data and navigation method. To that end, HDML has been designed with the "deck of cards" metaphor that are accessed through a transaction-based navigation model.

HDML uses a "card" as the basic element, with a number of cards -- a "deck" -- required to make up an application. This difference is more than a semantic difference from HTML "pages". Cards are individual units of data that are often, on some level, atomic -- unable to be broken up until smaller pieces without losing context. A deck of cards is linked internally and can be referenced by other decks and cards.

Because the card is designed to be discrete units of data, there are multiple types of cards: cards for displaying data, cards for entering data, and cards for listing indexes or lists of choices. Putting multiple types of data on a single card would imply the user has enough visual context in order to understand the differences. With the small display, the context would be lost.

This segregation of data motivates a navigation model that is transaction based, rather than the navigation model found on the web today, browsing or surfing. A deck of cards is designed to walk the user through a number of choices and data entry to result in a request which will return the specific data desired. The end result could be a stock quote, a weather report, or a phone number, with varying levels of input and selection required to arrive at that data. The transaction-based model allows a developer to present the user with an efficient method of selecting and receiving that data the user needs at that moment.

5.0 HDML 2.0 Specification

HDML was designed because no other markup or scripting language was available which met the needs of handheld devices with limited resources -- memory, processing speed, and bandwidth. HDML is currently in use by a number of service providers and device manufacturers. A user agent has been developed and ported to 10 different platforms, including Unix and Windows. Hundreds of applications are available today and thousands of developers have begun developing applications based on this initial user agent implementation.

The complete specification for HDML is available at <http://www.w3.org/pub/WWW/TR/NOTE-Submission-HDML-spec.html>

HDML 2.0 Proposal - 11 APR 1997

For more information about this proposal, contact:

[Tim Hyland](#)

Product Manager

Unwired Planet