Browsing Enhancements Phase One Requirements

Candidate Version 1.0 – 16 December 2003

Open Mobile Alliance

OMA-RD-BrowsingEnhancementsOne-V1_0-20031216-C

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the “OMA IPR Declarations” list at http://www.openmobilealliance.org/ipr.html. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR’S REPRESENTED ON THE “OMA IPR DECLARATIONS” LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.

THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.

Contents

1. Scope (Informative)

This document defines the requirements for the first phase of the Browsing Enhancements work item.

2. References

2.1 Normative References

[RFC2119] “Key words for use in RFCs to Indicate Requirement Levels”. S. Bradner. March 1997.
URL:http://www.ietf.org/rfc/rfc2119.txt

[XHTMLMP] “XHTML Mobile Profile”, Open Mobile Alliance™,
http://www.openmobilealliance.org/documents.html

[WCSS] “Wireless Cascading Style Sheets version 1.1”, Open Mobile Alliance™,
http://www.openmobilealliance.org/documents.html

[DRM] “Digital Rights Management version 1.0”, Open Mobile Alliance™, OMA-DRM-v1_0,
http://www.openmobilealliance.org/documents.html

2.2 Informative References

[XHTMLMOD] “Modularization of XHTML™”, W3C Recommendation 10 April 2001.
URL: http://www.w3.org/TR/xhtml-modularization/

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

None.

3.3 Abbreviations

WCSS	Wireless Cascading Style Sheets
XHTML	Extensible Hyper Text Markup Language
XHTMLMP	XHTML Mobile Profile
MMI	Man-Machine Interface
W3C	World Wide Web Consortium

4. Introduction

Browsing Enhancements is a general work item for new browser capabilities. The proposed enhancements in the first phase are the result of proposals from member companies to provide short term improvements in function and enhance usability and interoperability for browsing.

The areas being addressed by phase one are:

- **Navigation**. The intention is to provide enhanced usability by recommending the content author to use, and browser vendors to implement, certain standard [XHTML] features and [WCSS] features that make it easier for the user to access links contained in, and navigate around, the current document and the Web. The intention is also to define a way for the content author to control the use of soft keys, when such are available.

- **Multimedia**. The intention is to provide enhanced interoperability and function by defining how to add multimedia objects such as Java games and video clips to Web pages. The <object> element will be used, which is defined in the [XHTMLMOD] standard and is already part of XHTML Mobile Profile (XHTMLMP), but exactly how it shall be used for specific multimedia objects is yet undefined.

- **File upload and other XHTML Form features**. The intention is to provide this new feature to allow the uploading of files from a device to a Web Server in a consistent manner. This will be achieved by adding to the XHTML Mobile profile the standard [XHTML] file input control. Since [XHTMLMOD] is a collection of modules, the complete XHTML Forms module should be considered for inclusion into XHTML Mobile profile.

- **Input mode**. The intention is to provide enhanced usability by defining a way for the content author to set the rules for any acceptable input, thereby allowing the device to ensure the inputs meet the rules before the input is sent to the server. This will be achieved by setting the input mode of a text input control to a value that is appropriate for the kind of data that the user is supposed to enter, e.g. the input mode for digits is appropriate when user is supposed to enter a number.

5. Use Cases (Informative)

The following use cases have been defined for each of the features listed above. The list is not intended to be an exhaustive list of use cases but illustrative for the features.

5.1 Navigation

5.1.1 Short Description

An application wishes to provide the best experience possible for the user. To achieve this certain links are desired on all documents comprising the application such as “help”, “home”, “previous”, “next”, etc. Moreover is the document is substantially more than a single screen it is recognised that scrolling is less than desirable and so such features as “Page Down”, “Page Up” etc. are useful.

While some of these features may be left to the implementations and therefore not require any explicit control in the document there are foreseeable cases where the content or application author wishes to have consistent user experience withing the limits of various implementations which requires explicit control mechanisms. Specifically the Content Author may wish to specify additional descriptive information about links in a Web page: (a) the relationship between the page and the link; (b) a shortcut to access the link; and (c) presentational characteristics of the link. Subsequently, when the browser displays the page, it uses the information to make the links easier to access for the user.

5.1.2 Actors

- **User**: the User is the person that is using the browser to view Web pages.
- **Content Author (a.k.a Author)**: the Content Author is the person that is creating the Web pages.

5.1.2.1 Actor Specific Issues

- **User**: The User wishes a pleasing experience when accessing content. Excessive use of the navigational device (keys, pointing device etc.) will detract from this experience. Long documents/pages may be informative but without easy navigation through the document and to related documents in the application they can be detractors from a pleasing experience. Lists of links within or at the end of a document for subsequent naviation are also detractors from a pleasting experience.
- **Author**: The author wishes to provide a pleasing experience to the user but cannot easily do so today as content has to be tailored to each device if an optimal experience is required adding to deployment time and cost. The author desires a well known means to have navigational short cuts within the document and to other documents comprising the application.

5.1.2.2 Actor Specific Benefits

- **User**: Enhanced usability. Easier access to some links.
- **Author**: Can define information about links so that the browser can make them available to the user in additional ways that enhance usability.

5.1.3 Pre-conditions

The User is browsing the Web, i.e. is connected to the Internet and a web server hosting the content.

5.1.4 Post-conditions

The User has activated one of the links in a Web page and continues to browse the Web.

5.1.5 Normal Flow

1. Author creates a Web page. The author may use one or more of the following:

   a. Shortcut keys are defined for links that the user should be able to access in “one click”. It is intented that the HTML “accesskey” attribute is used to achieve this.

   b. Special markup is used for links that are used to navigate the Web site (start page, help page, main menu, next and previous page, etc.) and that does not belong to the content as such. It is intended that the HTML “rel” attribute and standard link types are used to achieve this.

   c. Presentation properties are defined for links that shall be presented as soft keys or as menu items, outside the normal presentation context of the page. It is intended that new properties in [WCSS] are used for this.

2. User navigates to the Web page.

3. The page is displayed.

   a. The shortcuts are made available to the user if used and supported. Links are assigned to shortcut keys according to the semantics of the “accesskey” attribute.

   b. It used and supported, the site navigation links are made available to the useroutside the normal presentation context of the page. The actual means of making such links available to the user may be implementation specific but will be consistent with the browser’s UI.

   c. It used and supported any links with presentational properties are presented according to the semantics of those properties.

5.1.6 Alternative Flow

Alternative flows are defined by the combinations of 1a, 1b, 1c.

The Author may use or skip any of these steps, or do all three.

5.1.7 Operational and Quality of Experience Requirements

The minimal number of links that the browser can display outside the normal presentation context must be specified.

5.2 Multimedia

NOTE: The MAE subworking group is looking into the consequences of the US Patent 5,838,906 before starting any work on this use case.

5.2.1 Short Description

A Content Author uses the <object> element to include a multimedia object into a Web page, such as a Java application, video clip, or sound file. During the rendering of the web page the object, which may be an executable file, is launched by the program capable of handling the object, such as a media player or Java machine. The Content Author may wish to define parameters in the Web page which are used to configure the program.

The <object> element in [XHTML] is defined to support this use case and will be used to meet the foreseeable use cases.

5.2.2 Actors

- **User** : the User is the person that is using the browser to view Web pages.

- **Content Author (a.k.a Author)** : the Content Author is the person that is creating the Web pages.

5.2.2.1 Actor Specific Issues

- **Author**: There is no standard way of defining how a multimedia object should be launched from a Web page, except that the <object> element can be used. A number of proprietary ways exists.

5.2.2.2 Actor Specific Benefits

- **User**: The user can launch multimedia objects directly from a Web page in addition to any that are automatically initiated when the document is rendered.
- **Author**: The author can define how the multimedia object shall be launched from the browser.

5.2.3 Pre-conditions

The User is browsing the Web, i.e. is connected to the Internet and a web server hosting the content.

5.2.4 Post-conditions

A multimedia object has been launched from the browser. The user continues browsing.

5.2.5 Normal Flow

1. The Author creates a Web page with a multimedia object. The object is included by reference from the <object> element. Parameters that control the launching of the object are defined in <param> element.
    a. A parameter that defines whether the object shall run inside the same presentational context as the current page, or in a new one. (In the same window or a new.)
    b. Parameters specific to each media type.
2. User navigates to the Web page.
3. The page is displayed and object is launched according to the parameters.
4. The user is interacting with the object; playing a game, listening to a sound file, etc.
5. The user stops interacting with the object and contintues browsing.

5.2.6 Alternative Flow

None have been defined.

5.2.7 Operational and Quality of Experience Requirements

If the MIME type of the object is not supported the standard fallback algorithm of the <object> element will take effect.

5.3 File Upload

5.3.1 Short Description

An application owner wishes to provide a service to a user whereby user provided information is made available to either the user for subsequent use or by others, e.g. personal web pages in a hosting environment. To achive this the author provides a web based authoring environment and the user sets the format of the pages and the textual content. The user may which to add content, e.g. a picture or other local file, from the device he/she is using to create the web pages and needs to upload the content to the web creation environment for inclusion in his/her web page.

The Content Author creates the means by creating a form in a Web page with the form control enabling the user to submit a binary file, located on the mobile phone, together with the rest of the form data to the server.

5.3.2 Actors

- **User**: the User is the person that is using the browser to view Web pages.
- **Content Author (a.k.a Author)**: the Content Author is the person that is creating the Web pages.

5.3.2.1 Actor Specific Issues

- **User**: Today the user may or may not have support for file uploading thereby limiting the applications and services available to him/her.

- **Author**: Today authors have varying levels of support for file uploading making the creation of services that could make use of uploaded content from users problematic if not impossible for many devices.

5.3.2.2 Actor Specific Benefits

- **User**: With this feature the user can upload files from the mobile phone to a Web server thereby enabling new classes of services to be used..

- **Author**: With this feature the author can create services such as Web-mail and Web-based picture galleries.

5.3.3 Pre-conditions

The User is browsing the Web, i.e. is connected to the Internet and a web server hosting the content.

5.3.4 Post-conditions

One or more binary files have been submitted to the server. The user continues browsing.

5.3.5 Normal Flow

1. Author creates a Web page which contains a form with an associated file input control.
    a. Author specifies, as an attribute of the input control, the expected MIME media type of the file.
2. User navigates to the Web page.
3. User selects the file input control.
4. A list of files available on the mobile phone is displayed to the user.
    a. The list of files can be filtered according to the MIME media types specified on the input control, i.e. if the MIME type is “image/gif”, only GIF images are displayed to the user.
5. User selects one or more files for upload and leaves the input control.
6. User submits the form to the server.

5.3.6 Alternative Flow

In 4a:

This step is skipped if no MIME media type is specified or if the browser does not support the specified type.

5.3.7 Operational and Quality of Experience Requirements

For files that are protected by DRM, the rights are taken into account. The user cannot upload a forward-locked file, for example.

When a file is uploaded it is copied, not moved. The file is in the same state after the upload operation, as it was before.

5.4 Input Modes

5.4.1 Short Description

A Content Author wishes to control a users inputted information into a webpage that will subsequently be submitted to the application server for processing.

The Content Author creates a form in a Web page with a text input control. For the text input control the Author specifies the appropriate input mode (predictive text, digits, etc.). When the User selects the text input control, the user interface is configured according to the specified input mode. This configuration acts only as the initial input mode and only for the input control that the Author has specified. The User is free to change the input mode at any time.

5.4.2 Actors

- **User**: The User is the person that is using the browser to view Web pages.
- **Content Author (a.k.a Author)**: The Content Author is the person that is creating the Web pages.

5.4.2.1 Actor Specific Issues

- **User**: The user gets frustrated when prompted to input a sequence of characters, e.g. indentifier, pressing the OK to submit the sequence, and subsequently being informed by the application to try again as it did not meet the criteria for the application after checking at the server. The user wishes to some intelligence to be applied to prompt locally upon submission.
- **Content Author (a.k.a. Author)**: The content author wishes to be able to predefine the allowable charaters for an input, e.g. indentifier, and know the device's user agent will manage initial checking to ensure they are of the right form, e.g. alpha, digits, etc.

5.4.2.2 Actor Specific Benefits

- **User**: Enhanced usability when entering text into form controls as the number of device >< server roundtrips is reduced, since initial checking is applied locally..
- **Author**: A way to specify the appropriate input mode for a form control, which enhances the usability of the application. The resulting improvements reduce checking and load on the server and roundtrips data transport costs.

5.4.3 Pre-conditions

The User is browsing the Web, i.e. is connected to the Internet and a web server hosting the content.

5.4.4 Post-conditions

Text has been entered into the text input control.

5.4.5 Normal Flow

1. Author creates a Web page with a text input control. The input mode for the text control is defined by the following variables:
   a. Whether predictive text is on or off.
   b. The type of characters, for example upper case, lower case, title case, digits.
   c. The Unicode character set.
2. User navigates to the Web page.
3. User selects the text input control.

4. The user interface is configured according to the defined input mode. For example if upper-case characters are defined then the user interface lets the user enter upper-case characters.

5. User submits the form to the server.

5.4.6 Alternative Flow

None has been defined.

5.4.7 Operational and Quality of Experience Requirements

The user can always enter text into the form control. If the user interface does not support the specified input mode, a default input mode is used.

Predictive text cannot be turned on if the user has explicitly turned it off in the settings for the mobile phone.

5.5 Open Issues

None have beed identified.

6. Requirements (Normative)

6.1 High-Level Functional Requirements

The following high level functional requirements are derived from the use cases specified in section 5.

6.1.1 Markup and Style Language

1. The markup language SHALL enable the author to specify “one click” shortcut keys.
2. The Markup Language SHALL enable the author to specify links to related resources on the Web, for example other resources on the same Web site as the current document.
   a. The <link> element [XHTMLMP] SHALL be used for this purpose.
3. The Markup Language SHALL enable the author to specify the type of a link, by using a well-known name for the type such as “start”, “prev”, and “next”.
   a. The “rel” attribute [XHTMLMP] SHALL be used for this purpose.
   b. The link type names defined in [XHTMLMOD] SHALL be used for this purpose.
4. The Style Sheet Language SHALL support properties to control the presentation of links outside the content area, for example as soft keys or menu items.
   a. Style properties in the “-wap”-namespace SHALL be used for this purpose.
   b. The style properties SHALL describe the presentation in terms of logical MMI mechanisms.
5. The Markup Language SHALL enable the author to add, by reference, multimedia objects.
   a. The <object> element [XHTMLMP] SHALL be used for this purpose.
   b. The <param> element [XHTMLMP] SHALL be used to control launching of the object. These parameters MAY be passed to the rendering or execution engine as appropriate.
   c. The Browser MAY display the multimedia object in the current context or in a new one. In a windows-environment, a window is a context.
   d. If the MIME media type of the object is not supported, the standard fallback algorithm defined for the <object> element SHALL apply.
   e. The solution SHALL be interoperable with existing implementations of the <object> element.
6. The Markup Language SHALL enable the author to create a file input control.
   a. The user SHALL be able to specify the file to be uploaded, by selecting the file from the list of files available on the device.
   b. The file selected by the user SHALL be uploaded to the location specified in the file input control.
   c. The uploaded file SHALL remain on the device; a copy of the file is uploaded.
7. The Markup Language SHALL enable the author to specify the input mode for text input controls.
   a. The input mode SHALL allow predictive key input to be enabled or disabled, provided the user has not explicity overridden this capability.
   b. The input mode SHALL allow the type of characters to be specified, such as “lower case”, “digits”, “letters”, and so on.

c. The input mode SHALL allow the Unicode charater set to be specified.

d. The input mode MAY allow other settings to be specified.

e. The input mode SHALL only act as the initial input mode for the text input control. The user SHALL be able to change the input mode at any time.

f. The input mode SHALL apply only on the input control for which it was specified. It SHALL NOT override user input preferences in general.

6.1.2 Browser (User Agent)

1. The Browser SHOULD make hyperlinks available outside the content area, for example in a menu, in addition to the standard presentation of links inside the content area. Hyperlinks are defined by the <a> element.
2. The Browser SHOULD make links defined by the <link> element and which have a well-known link type defined in the “rel” attribute, available to the user.
3. The Browser SHOULD use information about the type of the link when making the link available to the user.
4. When alternative ways of accessing a link in the page are available (e.g. an accesskey-shortcut), the Browser SHOULD provide a presentational hint about this to the user.

6.1.3 Security

1. When uploading files, the device SHALL NOT violate requirements specified in other enablers (e.g. DRM [DRM]) that the device supports.

6.1.4 Charging

No requirements.

6.1.5 Administration and configuration

No requirements.

6.1.6 Usability

All of the features stated in section 5 provide improvements in the usability of services exploiting them.

6.1.7 Interoperability

1. Parameters used in the <object> element to configure the launch of a multimedia object MUST be defined for common MIME media types.
2. The style properties, in the Style Sheet Language, for soft keys MUST be defined in a way that guarantees the same user interaction indepedant of the browser vendor.

6.1.8 Privacy

No requirements.

6.2 Overall System Requirements

6.2.1 Use of standards

1. Existing Web standards such as [XHTMLMP] and [WCSS] MUST be used wherever possible, instead of defining new technologies.

It is anticipated that the vast majority of the features defined in section 5 can be realised by appropriately using existing technologies and adding them to [XHTMLMP] and [WCSS].

6.3 System Elements

6.3.1 Network interfaces

1. The Markup Language, which is used between the Browser and the Server, MUST be [XHTMLMP].

2. The Style Sheet Language, if used, MUST be [WCSS].

Appendix A. Browser Enhancements (Informative)

This section describes the four main areas of enhancements.

Navigation

Links are navigation options to the user and should be easy to access at all times while viewing a Web page. Many Web pages contain lots of links to other parts of the same document, to other parts of the same Web site, and to other sites. On a PC, the browser is able to display an enough large portion of the Web page so that all important links are easy to find by the user, without the need to scroll down. On a small display, where only a small portion of the page can be displayed at the same time, the user cannot always see all links that are available in the page. The user may have to scroll to the bottom of the page before finding the links, the available navigation options from the page. If the links are placed at the top of the page, other even more important parts of the page may not be displayed without scrolling down the page.

The purpose of this enhancement is to make it easier for the user to access links in Web pages and navigate around the Web. There are two approaches to the solution.

- **Browser-centric**. Define a standard way for content authors to characterize the links in Web pages, and let the browser decide, based on the type of the links, how to best present them the user. The browser vendor is responsible for making sure that the links are presented in a way that is aligned with the rest of the MMI of the device; for example by presenting links in the same way as navigation options in general are presented in the MMI.

- **Content-centric**. Define a standard way for the content author to specify in the content how links shall be presented in the browser. The content author is responsible for making sure that the links are presented in a way that makes sense on the browser and with the MMI that the user is using to view the Web page. There may be a mismatch between how much of the presentation the content author is able to express in the content, and what the browser is able to do.

The above are not mutually exclusive.

Multimedia

Multimedia objects, including execuables such as Java applications, are embedded in Web pages by using the <object> element. The element is part of XHTML Mobile profile. In its most simple form, the element is a general form of the more specific <img> element, which embeds images only.

It has proved hard to fully implement the <object> element in a PC-browser, and make it interoperate with other browsers. In a mobile phone it is harder still. The <object> element has a large number of attributes, and semantics appropriate for PC-browsers. It also has a child-element, <param>, which can be used to define parameters for the object referenced by the parent <object> element. Because the actual usage of the <object> and <param> element has never been adequately defined in a standard, there are interoperability problems between how PC-browsers support the <object> element. There are for example proprietary solutions for how the <object> element is used to launch a media player from a Web page, depending on the platform and media player. Adding to the problem is the fact that the proprietary <applet> and <embed> elements are used for similar purposes

The purpose of this enhancement is to define how the attributes and child-elements of the <object> element are used to include multimedia objects into XHTML Mobile profile documents, and avoid the kind of interoperability problems that exist today between PC-browsers.

File Upload

The XHTML Mobile profile contains a subset of XHTML forms, which are used to create fill-in forms. One of the things left out is the ability to create a file input control. File input controls are used in for example Web-mail services to let the user add files as attachments to e-mails. When the user selects the control the browser lets the user select one or many files from the file system, which are then submitted to the server together with the rest of the form data.

One of the challenges of supporting file upload in a mobile phone is that there may not be a file system. Images, ring-tones, and other media objects, which of course are a kind of files, may be organized into folders and presented to the user as different menus, but there may not be the kind of file system users can see on a PC, it may look totally different on different mobile phones. This also means that what happens when the user selects the file input control may be totally different in different browsers, which is a potential interoperability problem.

File upload is one of several features in the XHTML Forms module (see XHTML Modularization [XHTMLMOD] for a list of all XHTML modules). The XHTML Mobile profile supports a defined subset of the Forms module, called the Basic Forms module, but lacks from the full Forms module, in addition to file upload, image maps and buttons. In order to conform to XHTML, a document type such as XHTML Mobile profile must support complete modules. This means that the full XHTML Forms module must be considered for the XHTML Mobile profile, in order to conform to XHTML.

The purpose of this enhancement is to add file upload to the XHTML mobile profile in a way that conforms to XHTML and that can be implemented in an interoperable way, despite the fact that mobile phones may not have files (as users know them on a PC).

Input Mode

The input mode may be a keyboard configuration (e.g. predictive text), an input method editor, or any other setting affecting input on the mobile phone. When used in the menu system the MMI always knows which one is the appropriate input mode, because it has been programmed that way (digits for phone numbers, etc.). But when the browser displays a text input control in a Web page, it cannot know what the appropriate input mode is. So browsers use the same input mode for all text input controls. As a result, the input mode is sometimes inappropriate for the kind of data the user is supposed to enter; for example predictive text when the user tries to enter a phone number.

The content author who has created the Web page, however, knows the appropriate input mode for every text input control. If the user is supposed to enter a number, the appropriate mode is digits; if the user is supposd to enter a name, predictive text may be the most appropriate mode; and so on.

The purpose of this enhancement is to define a way for content authors to set the input mode for text input controls in Web pages.

The WCSS '-wap-input-format' property can be used to define the format (using a trivial kind of regular expressions) of the data the user is expected to enter. The format gives the browser a hint of which one is the best input mode. But an indirect hint is not as good as an explicit directive, but clearly the relation between the two needs to be defined.

Some users don't like predictive text systems and turn that feature off. This group of users will not be helped if the content author turns predictive text on again. User preferences must be weighted against author preferences.

As part of the XForms activity, W3C has defined the "input-mode" attribute, and a large number of possible values, which is intended to solve this problem. The specification of the attribute is in appendix E of the XForms specification, <http://www.w3.org/TR/2002/CR-xforms-20021112/sliceE.html>. The attribute is not part of XHTML, but since it is a W3C recommendation, it should be taken into account before another solution is though

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description

A.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Versions OMA-RD_BrowsingEnhancmentsOne-V1_0	2003-10-02	5.1.7, 6.1.3, 5.4.1	Updated after REQ review. See report in OMA-REQ-2003-0593.
	2003-10-08	6.1.2	Added requirement #4. See report in OMA-REQ-2003-0593.
Candidate Version OMA-RD-BrowsingEnhancementsOne-V1_0	16 Dec 2003	n/a	Status changed to Candidate by TP TP ref # OMA-TP-2003-0643-Request-approval-Browser-enhancement-req