METHOD AND APPARATUS FOR AUTHORING AND PUBLISHING WEB PAGES

Field of the Invention

The present invention relates to authoring web pages for publication on the World Wide Web, and more particularly for authoring web pages that can be presented on a variety of web enabled devices in accordance with an author's web page design.

Background of the Invention

The Internet and more particularly the World Wide Web (WWW) has redefined publication and distribution of information. Initially computers, such as desk-top computers and portable computers, were able to connect to and access the Internet, and typically the connections were provided by wired local area networks or dial-up modems, and telephone lines. Connections to the Internet is now available through wireless means, such as through broadband wireless networks or via a cellular telephone service provider, and with the availability of wireless Internet access, a variety of portable devices that have in-built Internet access have become available.

These portable devices take many forms. For example, cellular phones have been enhanced to support Internet access, and there are also personal digital assistants that have integrated Internet access capabilities. Unlike computers that have large displays for presenting the web pages of the WWW, the portable devices have relatively smaller displays, typically liquid crystal based, which have significantly limited display capabilities. In addition, as there is no standardization of the type of displays, presenting the web pages of the WWW on the various display types with a consistent appearance poses much difficulty.

An initial approach was for an author of a web page to author different markup versions of a web page, each markup version being suitably coded for display on a particular device. Then, in operation, when a particular device sends a web page request to a server that hosts the various markup versions of the requested web page, the server determines the type of device from the web page request, gets the corresponding particular version of the requested web page from server memory, and sends the particular version of the web page to the particular device. Since the version of the web page that is received by the particular device was designed by the author specifically for that device, the received version of the web page is displayed as designed by the author.

As will be appreciated, this method is inefficient and it amounts to multiplying the author work by the number of devices that are to be supported. In addition, when new devices are to be supported, the author has to prepare the respective versions of existing web pages for the new device. Further, as the author has to manually design each of the various versions of the web pages within the constraints of each of the devices, the various versions of a web page may not have a consistent appearance due to design choices of the author and/or limitations of the particular device.

International Business Machines (IBM) of the USA provides an alternative approach, with a product known as Websphere™. The IBM method provides a proxy server with a transcoder that is linked to the website where an author hosts his web pages. When a user of a portable device requests one of the web pages, the request is sent to the proxy server, where the transcoder retrieves the requested web page from the authors web site. Then, after identifying the type of portable device from the request, the transcoder automatically generates a version of the requested web page using the retrieved web page, where the generated web page is suitably adapted for display on the

requesting portable device. The proxy sever then sends the adapted web page to the portable device.

With the proxy server method, the author of a web page need not generate multiple versions of each web page. However, a disadvantage of the proxy server method is that the author has limited control of the appearance of the various versions of his web page when generated for, and displayed on, a variety of portable devices. This is because the proxy server automatically generates the various versions of the web pages for the variety of devices.

Another disadvantage of the proxy server method is the need for a user of a device to select the Websphere proxy server as the proxy server for the device. This can be an inconvenience for the user, for example, when the user subscribes to a service provider having a proxy server that does not support Websphere.

Morphis is an open-source wireless transcoding platform by Kargo Inc. of the USA. Morphis includes a set of tools and an abstract markup language called Wireless Abstract XML (WAX), which is used to author content for wireless applications. Morphis uses extensible style sheet language (XSL) style sheets to translate a web page authored in WAX language to hypertext mark-up language (HTML), handheld device mark-up language (HDML), wireless mark-up language (WML), or whatever language that can be intepreted by a requesting device. The XSL style sheets require XSL tranformations (XSLT) to perform the tranlsations from WAX, and it is the XSL transformations that provide layout information, much like a template, for displaying a WAX authored web page in accordance with an author's design. Using XSLT, an author has more control of the appearance of a web page as it appears on each of the variety of devices. However, a disadvantage of the Morphis transcoding platform is the need for an author to create separate XSLTs for each new web page layout, and also for each of the variety of devices that are to be provided access to the WAX authored web pages.

Brief Summary of the Invention

The present invention seeks to provide a method and apparatus for authoring and publishing web pages, which overcomes or at least reduces the abovementioned problems of the prior art.

Accordingly, in one aspect, the present invention provides a transcodable formatting language for specifying the format of an electronic document, wherein the transcodable formatting language is transcodable to at least one presentation language, the at least one presentation language for presenting a transcoded version of the electronic document on at least one of a variety of devices, the transcodable formatting language comprising:

at least one author-specific display-component that is presented on the at least one of the variety of devices in accordance with author specifications; and

at least one device-specific display component that is presented on the at least one of the variety of devices in accordance with device specifications of the at least one of the variety of devices.

In another aspect the present invention provides an authoring tool for authoring an electronic document in a transcodable formatting language, wherein the transcodable formatting language is transcodable to at least one presentation language, the at least one presentation language for presenting a transcoded version of the electronic document on at least one of a variety of devices, the authoring tool comprising:

a first input for receiving at least one display-component from an author of the electronic document;

a second input for receiving layout information associated with the at least one display-component from the author of the electronic document;

a third input for receiving display-component specification associated with the at least one display-component from the author of

the electronic document, the display-component specification for indicating the at least one display-component will be presented in accordance with specification provided by the author when the display-component specification is author-specific, and the display-component specification indicating the at least one display-component will be presented in accordance with device specification of the at least one of the variety of devices when the display-component specification is device-specific; and

an output for providing the electronic document in the transcodable formatting language.

In yet another aspect the present invention provides a method for authoring an electronic document in a transcodable formatting language, wherein the transcodable formatting language is transcodable to at least one presentation language, the at least one presentation language for presenting a transcoded version of the electronic document on at least one of a variety of devices, the method comprising the steps of:

a) providing at least one display-component;

b) providing display-component specification for the at least one display-component;

c) providing layout information for the at least one display-component; and

d) generating the electronic document in transcodable formatting language, wherein the electronic document includes the at least one display component, the display-component specification associated with the at least one display-component, and the layout information associated with the at least one display component.

In still another aspect the present invention provides a transcoding system for transcoding an electronic document to at least one presentation language, the at least one presentation language for presenting a transcoded version of the electronic document on at least one of a variety of devices, the transcoding system comprising:

a first input for receiving the electronic document in response to a request from the one of the variety of devices, wherein the electronic document is authored in a transcodable formatting language;

a second input for receiving a device profile of the one of the variety of devices;

a processor coupled to the first and second input, the processor for determining the device profile of the one of the variety of devices is associated with the at least one presentation language, and the processor for processing the electronic document to produce the transcoded version of the electronic document in the at least one presentation language; and

an output for providing the transcoded version of the electronic document in the at least one presentation language to the one of the variety of devices.

A further aspect of the present invention provides a method for transcoding an electronic document in a transcodable formatting language, wherein the transcodable formatting language is transcodable to at least one presentation language, the at least one presentation language for presenting a transcoded version of the electronic document on at least one of a variety of devices, the method comprising the steps of:

a) determining a request from the at least one of the variety of devices has been received;

b) determining from the request that the at least one of the variety of devices presents the transcoded version of the electronic document in the at least one presentation language;

c) transcoding the electronic document in the transcodable formatting language to the transcoded version of the electronic document in the at least one presentation language; and

d) providing the transcoded version of the electronic document to the at least one of the variety of devices for presentation thereon.

Another further aspect of the present invention provides an authoring tool service provider for providing an authoring service to at least one user, the authoring tool service provider for authoring an electronic document in a transcodable formatting language, wherein the transcodable formatting language is transcodable to at least one presentation language, the at least one presentation language for presenting a transcoded version of the electronic document on at least one of a variety of devices, the authoring tool service provider comprising:

an input coupled to receive at least one display-component, layout information for the at least one display-component, and display-component specification associated with at least one display-component of the electronic document, the display-component specification for indicating the at least one display-component will be presented in accordance with specification provided by the at least one user when the display-component specification is author-specific, and the display-component specification indicating the at least one display-component will be presented in accordance with device specification of the at least one of the variety of devices when the display-component specification is device-specific; and

an output adapted to provide the electronic document in the transcodable formatting language to the at least one user.

Another still further aspect of the present invention provides a transcoding service provider for providing a transcoding service to at least one subscriber, the transcoding service provider for transcoding an electronic document to at least one presentation language, the at least one presentation language for presenting a transcoded version of the electronic document on at least one of a variety of devices, the transcoding service provider comprising:

an input for receiving the electronic document, wherein the electronic document is authored in a transcodable formatting language, and for receiving a device profile of the one of the variety of devices; and

a processor coupled to the input, the processor for determining the device profile of the one of the variety of devices is associated with the at least one presentation language, and the processor for processing the electronic document to produce the transcoded version of the electronic document in the at least one presentation language; and

an output for providing the transcoded version of the electronic document in the at least one presentation language to the at least one subscriber.

Brief Description of the Drawings

An embodiment of the present invention will now be more fully described, by way of example, with reference to the drawings of which:

FIG. 1 shows a functional block diagram of an authoring tool in accordance with the present invention;

FIG. 2 shows a graphical user interface of the authoring tool in FIG. 1;

FIG. 3 shows a flowchart detailing the operation of the authoring tool in FIG. 1;

FIG. 4 shows a functional block diagram of a transcoder in accordance with the present invention;

FIG. 5 shows a flowchart detailing the operation of the transcoder in FIG. 4;

FIG. 6 shows a functional block diagram of a system for providing authoring tool service in accordance with the present invention;

FIG. 7 shows a flowchart detailing the operation of the system in FIG. 6;

FIG. 8 shows a functional block diagram of a system for providing transcoding service in accordance with the present invention;

FIG. 9 shows a flowchart detailing the operation of the system in FIG. 8; and

FIGS. 10A-10F shows a variety of screen shots on portable devices that present web pages transcoded in accordance with the present invention.

Detail Description of the Drawings

The present invention provides a web page authoring tool to which a web page author provides web page content comprising various components for display i.e. display-components, which includes text and images. In addition, for each of the display-components, the author can set each selected display-component to be either author-specific i.e. displayed on a web page in accordance with the author's specification, or device-specific i.e. displayed on a web page in accordance with the device's specification. The author also provides layout information for each of the various display-components to the authoring tool, where the layout information indicates the position of a particular display-component relative to other display-components, on the web page being authored. The authoring tool then generates an iFormat markup language (IFML) file for the display-components and includes the layout information for each of the display-components. The IFML file is then stored on a server along with image files of display–components that are images.

When the server receives a web page request from a user device for the IFML authored web page, a request processor determines from the web page request the type of requesting device from a device profile memory, and hence the web page display capabilities of the requesting device are determined. Based on the web page display capabilities of the requesting device, a markup language that is compatible with the requesting device is selected. An IFML transcoder, associated with the server, then transcodes the IFML file into the selected markup language. The transcoder uses the layout information of the display-components in the IFML file to generate the transcoded web page in the

selected markup language, such that the relative positions of the author –specific display-components and device-specific display-components, as specified by the author, will be substantially preserved, where this is possible.

The transcoding process also includes optimizing the images from the image files to provide optimized images, which can be accommodated into the transcoded web page. The selected markup language can, for example, include hypertext markup language (HTML) or wireless markup language (WML), which can be interpreted or decoded by the requesting device. The server then sends the transcoded web page in the selected markup language, along with the optimized images, to the requesting device for presentation.

The present invention advantageously allows a web page author to design a web page once, and have the web page provided for display on a variety of devices with a variety of display capabilities, automatically, without the need for a proxy server, and substantially in accordance with the author's web page design.

Also, as the author-specific display-components are displayed consistently on the variety of devices, and the device-specific display-components are displayed on each of the variety of devices in accordance with the particular specifications of the particular device, the author has control over which display-components will be displayed consistently, and which will be displayed in accordance with the device specifications.

Another advantage of the present invention, as described, is that the author does not need to design additional web pages, or provide additional templates or formatting information, whenever a new web page layout is required.

The term "author" as used herein refers to the common meaning of the term in relation to web pages i.e. one who writes or constructs a web page or a system, such as a website.

The term web page refers to an electronic document or a document in electronic form available or published on the World Wide Web of the Internet. Typically, a web page is authored in markup language by an author, who is sometimes referred to as a web page designer. The markup language representing the web page, along with any image files, are communicated over the Internet; and, in a computing device having a display, the markup language of the web page is received and processed by a software application known as a Web Browser. The Browser renders or presents the web page, including the images, on the display in accordance with a layout, which is specified by the mark-up language of the web page.

With reference to FIG. 1, an IFML authoring tool 100, in accordance with the present invention, is used by an author to design one or more web pages. The authoring tool 100 comprises an input 105 for receiving content for display on a web page (not shown). The content comprises display-components 110, which include textual components 115 and image components 120, and are typically provided in a content file. The IFML authoring tool 100 has second input 125 for receiving layout information 130 for each of the display-components 110. The layout information 130 indicates the relative position of each of the display-components 110 on a web page that is being authored, and the layout information associated with each of the display-components 110 is provided directly to the authoring tool 100 by an author of the web page. The author can provide the relative positions of the display components graphically, such as by dragging icons of the display components within a window representing the web page being authored. Alternatively, the author can provide the relative positions using numerical entry of, for example, row and column numbers, representing a coordinate system for a display.

The IFML authoring tool 100 has a third input 132 for receiving display-component specifications 134 from the author. For example, the display-component specification 134 for a particular display-component

110 can be set by the author to be author-specific or device-specific. An author-specific display-component is one which the author wants displayed in a consistent manner on any device, while a device-specific display-component is one where the author wants, or allows, the display–component to be displayed in accordance with the particular specifications of any of a variety of devices.

Using the display-components 110, the display-component specifications 134, and the associated layout information 130, all of which are provided by the author, the IFML authoring tool 100 generates and provides an IFML file 135 via an output 140. Hence, the IFML file advantageously includes the display-components, the display-component specifications, and the information indicating the relative positions of the display-components on a web page.

In one embodiment, the IFML authoring tool 100 comprises a Java application program, which is executable using Java Runtime Environment (JRE) version 1.2 or later. Generic information in relation to Java Programming, Java application programs, and Java Runtime Environments, as enabled on personal computers and the like, will be known to one skilled in the art, and no further elaboration is provided herein, except where such elaboration will facilitate an understanding of the description of the embodiments of the present invention, as described herein.

Referring now to FIG. 2, a graphical user interface (GUI) 200 for an embodiment of the authoring tool 100, comprises a display-component listing area 205, which lists all the display-components, both textual and images, that may be selected to be displayed on one or more web pages. The textual components include basic text components, form components for presenting user-fillable forms on a web page, and dynamic components for running programs, typically Java software programs known as Java beans, on a web page.

The image components include images in a variety of image file formats, such as Joint Photographic Experts Group (JPEG), Graphics

Interchange Format (GIF), Tagged Image File Format (TIFF), Portable Network Graphics format (PNG), and Wireless Bitmap format (WBMP).

Prior to activating the authoring tool 100, an author specifies the display-components that are to be displayed on one or more web pages. This is accomplished with an XML content file, and an example of such an XML content file that specifies textual and image components follow.

```
<xml title="XML in 10 points">
  <graphic id="1">
    <image type="WBMP" src="w3c_main.wbmp"/>
    <image type="JPEG" src="w3c_main.jpg"/>
  </graphic>
  <point id="1">
    <title>XML is a method for putting structured data in a text
      file</title>
    <description>For "structured data" think of such ...
      platform-dependency.</description>
  </point>
```

The GUI 200 of the authoring tool 100 includes a content file area 210, an associated browse button 215 and a file open button 220, for the author to locate the XML content file. Author-specific select button 230, device-specific select button 235, and display-component deselect button 240, allow an author to select display-components from the display-component listing area 205, and set whether the selected display-component is to be author-specific or device-specific. It will be appreciated by one skilled in the art the number and arrangement of buttons can be varied and more buttons may be required when more complex intelligent functionality are incorporate in the authoring tool.

The selected display-components appear in a display-component positioning area 245. The author then has the option of moving the display-components within the display-component positioning area 245,

to thereby arrange the selected display-components in accordance with his preferred web page design. When arranging the display-components the author edits the display-component in the display-component positioning area 245, for example, editing here can comprise, the author adding or changing program codes in the display-component positioning area 245. In the editing process, the authoring tool 100 uses its in-built intelligence, enabled using XML schemas technology, to ensure the authoring complies with predetermined IFML grammar.

The display-component positioning area 245 has a co-ordinate system, which may be specified in rows and columns. Positioning the display-components in the display-component positioning area 245 associates each of the display components with a particular row and a particular column. An output IFML button 250 when activated by the author causes the IFML file 135 to be generated, and provided via output 140 of the authoring tool 100. The generated IFML file 135 can be used to generate markup language of a web page for a particular device, where the device presents the web with the display-components substantially as arranged in the display-component positioning area 245.

With reference to the flowchart in FIG. 3, the operation 300 of the authoring tool 100 when authoring a web page, starts 305 with identifying 310 the content file, which defines the display-components. Of course the content file in XML would have to be prepared in advance, prior to using the authoring tool 100. To identify a content file to the authoring tool 100, the author navigates the authoring tool 100 to locate the content file using the browse button 215; and when the content file is located, the path to the content file is displayed in the content file area 210. The file open button 220 is then activated by the author, and the display-components defined in the selected content file are listed 315 in the display-component listing area 205.

Next, the authoring tool 100 determines 320 whether the author has selected one of the listed display-components, and if so, a further

determination 325 is made as to whether the author has activated the author-specific button 230. If so, then the authoring tool 100 displays 330 the selected display-component in the display-components positioning area 245 as an author-specific display component. When the author-specific button 230 is not activated by the author, the authoring tool 100 determines whether the author has activated the display-specific button 235, and if so then the authoring tool 100 displays 340 the selected display-component in the display-components positioning area 245 as a device-specific display component.

Next, the author arranges 345 the displayed author or device specific display component in the display components positioning area 245. In this way the author provides the layout information 130 to the authoring tool 100. The authoring tool 100 then determines 350 whether the author has selected another display-component in the display-components listing area 205, and when another display-component has been selected, the process 300 returns to step 325 and proceeds as described earlier.

However, when the authoring tool 100 determines that the author has activated the IFML output button 250, the authoring tool 100 generates 360 the IFML file 135 that includes the display-components and the relative positions of the display-components in the display-component positioning area 245, and the process 300 then ends 365. However, when the IFML output button 250 is not activated by the author, the process 300 returns to step 350 of determining whether the author has selected another display-component in the display components listing area 205, and the process 300 proceeds as previously described.

In short, the authoring tool 100 described above allows an author to attach "device dependant" or "device independent" display commands to each display-component and finally generates the IFML for the whole web page designed by the author.

With the authoring tool 100, the author may also author web pages using templates. This applies typically to portable digital assistant (PD) type devices, where display capability is a lot more powerful than WAP-devices, but yet a lot lower than desk-top PC, and the display capabilities among various PDAs varies substantially. An author can select from predetermined templates, which are then presented in the display-components positioning area 245. For example, a menu bar can be provided for the author to select from various templates. A template defines patterns of display areas in the display-components positioning area 245. During the authoring process, the author allocates the display-components to the pre-defined display areas, as defined by a particular selected template. In addition, the authoring tool 100 includes a database, which defines the allowable user-agents, as defined by browser applications on user devices, and detectable through hypertext transfer protocol (http) and wireless application protocols (WAP) for each template. After composing a web page with a specific template, the author can then define the user-agents (browsers) applicable for that composed web page. The end result is that, by authoring through the use of templates, the author could effectively author once and cover a range of mobile devices instead of authoring for each of them separately.

Hence, the authoring tool in accordance with the present invention, advantageously allows an author to specify the display-components that are to be included in a new web page design, to set the display-components specifications for each of the selected display-components, and specify the layout of the display-components for a new web page.

With reference now to FIG. 4 a transcoding system 400 on a server that is coupled to the Internet, comprises a memory 405 coupled to an IFML processor 410. The memory 405 has four portions: a device profiles portion 412 for storing predetermined profiles of a variety of devices that may request a web page; an IFML files portion 414 that

represent web pages, such as those generated using the authoring tool 100; an images portion 416 for storing the images associated with the IFML files; and a Java beans portion 418 for storing Java programs that provide dynamic content for a web page.

A device profile of a particular device that is stored in the device profiles portion 412 comprises a user agent that uniquely identifies the particular type of device, and the display capabilities or specifications of the particular type of device. The display capabilities indicate the type of browser application that is used by the particular device, and hence the markup language for web pages that are to be presented on the particular device.

The memory 405 is also coupled to a device profile administrator 420. The profile administrator 420 provides a user interface (not shown) to allow device profiles that are stored in the device profiles portion 412 of the memory 405, to be amended. In addition, new device profiles can be added and outdated device profiles can be removed, using the device profile manager 420. Like the IFML processor 410, the device profile administrator 420 comprises a Java application program, and in one embodiment the device profiles are defined in an XML document, which can be edited by the device profile administrator 420. The use of XML allows the device profiles portion 412 to be scalable.

A request processor 425 receives web page requests 430 via input 427, from a variety of devices (not shown). A web page request 430 from a requesting device includes a user agent 432 and a universal resource locator (URL) 434, which is more commonly referred to as the web address of the requested web page. The user agent 432 indicates the type of device that sent the web page request 430. The request processor 425 is also coupled to the device profiles portion 412, and when the request processor 425 receives the web page request 430, the request processor 425 determines the user agent 432 therefrom, and accesses the device profiles portion 412 to determine the corresponding device profile of the requesting device. The request processor 425 then

provides the corresponding device profile to the IFML processor 410. In addition, the request processor 425 also determines the universal resource locator (URL) 434 of the requested web page from the web page request 430, and provides this URL 434 to the IFML processor 410.

The IFML processor 410 uses the URL 434 of the requested web page and retrieves a corresponding IFML file for the requested web page from the IFML files portion 414. Based on the device profile provided by the request processor 425, and using the retrieved IFML file and any associated images from the images portions 416, the IFML processor 410 generates and provides a transcoded web page 435 in the requesting device's markup language via output 440, to the requesting device. The transcoded web page 435 also includes the optimized versions of the associated images.

When generating the transcoded web page 435, the IFML processor 410 transcodes author-specific display components in the IFML file into the requesting device's markup language such that the author-specific display-components are displayed in accordance with the author's specifications. Alternatively, the IFML processor 410 transcodes device-specific display components in the IFML file into the requesting device's markup language such that the device-specific display-components are displayed in accordance with the device's specifications. For example, an author may want most of the textual display-components to be author-specific, and the image display-components to be device-specific, in order to ensure that whichever device request's a web page, the web page that is provided will have most of the textual information.

In one embodiment, the transcoding system 400 comprises a number of Java application programs that are deployable on any Java enabled web server, and the IFML processor 410 comprises one of the Java application programs that is coupled to access the IFML files portion 414, the images portion 416, and the Java beans portion 418, of the memory 405.

Referring to FIG. 5 the operation 500 of the transcoding system 400 on the server starts 505 with the request processor 425 determining 510 whether the web page request 430 has been received. When the request processor 425 receives the web page request 430 from any one of a variety of devices (not shown) via the Internet, the request processor 425 processes the web page request 430 to determine 515 the user agent 432. The request processor 425 then uses the user agent 432, accesses the device profiles portion 412, determines 520 the device profile of the requesting device therefrom, and provides the device profile to the IFML processor 410. Subsequently, the IFML processor 410 determines 525 the markup language of the requesting device from the device profile.

The request processor 425 also processes the web page request 430 to determine 530 the URL 434 from the web page request 430, and then provides the URL 434 to the IFML processor 410. Next, the IFML processor 410 uses the URL 434 to retrieve 535 a corresponding IFML file from the IFML files portion 414, and transcodes 540 the IFML file to the markup language of the requesting device, thus generating the requested web page in the markup language of the requesting device. During transcoding the author-specific display-components are coded in the markup language of the requesting device to be presented in accordance with the author specifications, and the device-specific display components are coded in the markup language of the requesting device to be presented in accordance with the requesting device's specifications. The IFML processor 410 accomplishes the coding by recognizing the author-specific and device-specific display-components in the IFML files, and generating the respective code in the markup language of the requesting device.

Also, during the transcoding process, any associated images identified in the IFML file are retrieved from the images portion 416 and optimized. Optimization comprises changes image files retrieved from the images portion into a format that is supported by the requesting

device. In addition, optimization can include reducing the size of the image files to be accommodated within the limited memory capacity of the requesting device. The requested web page in the markup language of the requesting device, including the optimized images, are then provided 545 to the requesting device. The process 500 then returns to step 510 and proceeds as described above.

Hence, the present invention as described, can advantageously transcode one IFML file associated with a predetermined URL to any one of a variety of corresponding web pages marked up in a one of a corresponding variety of markup languages. In addition, this can be accomplished automatically, without the need for the author of the IFML file to provide any further substantive input.

In another embodiment, the functionality of the authoring tool may be provided as a service by an authoring tool service provider via the Internet, and a description of a system for providing an authoring tool service now follows.

With reference to FIGS. 6 and 7, an authoring service provider 605 comprises a server coupled to the Internet 610. The server hosts an authoring tool service 615, which may be enabled by a Java application program on the server. Of course in the order to operate as a service provider the server 605 may also include registration services for users/subscribers of the service, and where the service is provided for a fee, the appropriate support to track usage and support billing and payment.

A user-terminal 620 typically comprises a computer 625 coupled to the Internet 610, and equipped with an appropriate operating system and browser application program. The operation 700 of a system for providing the authoring tool service starts 705, when an author at the user-terminal 620, wishes to use the authoring tool 615. The user-terminal 620 accesses 710 the server of the service provider 605 via the Internet 610 using the browser application and, where required, the author logs in as a registered user. In response, the user-terminal 620

receives 715 the user interface 200 from the authoring tool 615. Here, the user interface 200 comprises a web page and consequently, it is provided in a compatible markup language to the browser application on the computer 625, which presents the user interface 200 on a display of the user computer.

Prior to accessing the authoring tool service of the service provider 605, the content file that specifies the display-components 110, is prepared. The content file includes author-specific and device-specific display-components that are to be included on one or more web pages to be authored by the author on the user-terminal 620. As described earlier, the content file can comprise an XML content file that specifies the display-components 110, which includes the textual components 115 and the image components 120.

Using the user interface 200, the author then provides 715 input to the user interface 200 in accordance with the author's web page design. To provide the input, the author begins with identifying the content file using the browse button 215 and the file open button 220 of the user interface 200, and when the content file is located, the path to the content file is displayed in the content file area 210. The author then proceeds to sequentially select display-components 110 in the display-components listing area 205 using the author-specific and the device specific buttons 230 and 235, and then arranging the selected author-specific and device-specific display-components in the display-components positioning area 245. The process of selecting and arranging display components is repeated until the web page is completed.

When the author activates the output IFML button 250, the user-terminal 620 then provides the author-specific and device-specific display components, and layout information of the author-specific and device-specific display components, to the authoring tool service provider 605.

The authoring tool 615 then uses 720 the selected the author-specific and device-specific display components and the layout information to produce an IFML file associated with the designed web page, and provides the IFML file to the user-terminal 620. When the user-terminal 620 receives 725 the IFML file, the author can cause the user-terminal 620 to provide the IFML file, and image files associated with the IFML file, to a server 630 that hosts the author's website.

The server 630 stores the IFML file in an IFML files portion 632 of server memory (not shown), similarly the image files are stored in a images portion 634 of server memory. The server 630 includes a transcoding system 636, to process requests for web pages and provide transcoded web pages, as described earlier. The process 700 then ends 730.

The transcoding system 636 is substantially similar to the transcoding system 400 described earlier, and adapted for operation on the server 630. In addition, provisions are included to manage the receipt and storage of IFML files in the IFML portion 632 and image files in the images portion 634.

Hence, the present invention, as described provides an authoring tool service that can advantageously be delivered over the Internet, to author one IFML file, which can then be used to generate and provide a variety of web pages using a corresponding variety of markup languages.

The functionality of the transcoding system may also be provided as a service by a service provider via the Internet, and a description of a system for providing a transcoding service now follows.

With reference to FIGS. 8 and 9, a transcoding service provider 805 comprises a computer system programmed to be a server, which is coupled to the Internet 610. The server 805 hosts an IFML processor 815, which may be enabled by a Java application program on the server. In order to operate as a service provider, the server 805 may also include registration services for users/subscribers of the service, and

where the service is provided for a fee, the appropriate support to track usage and support billing and payment.

A user-server 820 comprises a server, which typically is a computer system, coupled to the Internet 610, that hosts one or more websites, and is a user of, or subscribes to, the transcoding service provider 805. The user-server 820 includes a request manager 825, and a device profiles portion 830, an IFML files portion 835, an images portion 840, and a Java beans portion 845. All of these portions are coupled to the request manager 825, and all these portions are stored in a memory (not shown) of the computer system.

The device profiles portion 830 stores device profiles of a variety of devices, the IFML files portion 835 stores IFML files that can be transcoded to web pages in markup languages of the variety of devices, the images portion 840 stores images associated with the IFML files in the IFML files portion 835, and the Java beans portion 845 stores Java programs that provide dynamic content for a web page.

The operation 900 of a system for providing the transcoding service starts 905, with the request manager 825 determining 910 whether the web page request 430 has been received via the Internet 610 from any one of a variety of devices. Upon receiving the web page request 430, the request manager 825 determines 915 the user agent 432 from the web page request 430, accesses the device profiles portion 830, and determines 920 the device profile of the requesting device (not shown) from the device profiles portion 830 using the user agent 432.

In addition, the request manager 825 also determines 925 the URL 434 of the requested web page from the web page request 430, and using at least a portion of the URL 434, retrieves 930 a corresponding IFML file from the IFML files portion 835 of the requested web page, along with any image files associated with the retrieved IFML file, from the images portion 840. Next, the request manager 825 provides 935 the device profile obtained from the device profiles portion 830, the IFML file retrieved from the IFML files portion 835, and any associated

image files of the IFML file, retrieved from the images portion 840, to the transcoding service provider 805.

The IFML processor 815 then determines 940 the markup language of the requesting device from the device profile, and based on the device profile provided by the request processor 825, and the IFML file and image files, the IFML processor 815 transcodes 945 the IFML to produce the requested web page in the requesting device's markup language.

The requested web page produced in step 945 also includes the optimized images, and where Java beans had been included with the IFML file, the requested web page includes dynamic content derived from the Java beans. Also, when generating the transcoded web page in step 945, the IFML processor 815 transcodes author-specific display components in the IFML file into the requesting device's markup language, such that the author-specific display-components are displayed in accordance with the author's specifications. Alternatively, the IFML processor 410 transcodes device-specific display components in the IFML file into the requesting device's markup language such that the device-specific display-components are displayed in accordance with the device's specifications.

The transcoding service provider 805 then provides 950 the markup language for producing the requested web page to the user-server 820. Upon receiving the markup language for producing the requested web page and the optimized image files 435, the request manager 825 provides 955 the markup language for producing the requested web page, along with the optimized image files 435, to the requesting device via the Internet 610. The process 900 then returns to step 910, and proceeds as described above.

In one embodiment, the transcoding service provider 805 comprises a number of Java application programs that are deployable on any Java enabled server, and the IFML processor 815 comprises one of the Java application programs.

Variations to the system for providing the transcoding service can include the user 820 further providing the address of the requesting device to the transcoding service provider 805, and the transcoding service provider 805 sending the markup language for producing the requested web page, and any associated optimized images, directly to the requesting device via the Internet 610.

A guide to IFML now follows, with TABLE 1 below providing the tags and attributes.

Body	Bgcolor Left Top	**Jsp**	
		LineBreak	
		Link	href
Column	Align Bgcolor Colspan Height rowspan Valign Width	**Nbsp**	
		Option	value
		Row	align bgcolor colspan height rowspan valign width
Form	Action method Name	**Select**	name
Group	Name	**Table**	align background border cellpadding cellspacing columns height width
Head			
Ifml			
Image	Alt border height hspace Src wapsrc Width	**Title**	
		WapBreak	
Input	Name Type Value		

TABLE 1

The general syntax of IFML is described below, and it should be noted that a complete IFML document must have a minimum set of tags, as illustrated in the following example of a minimum parsable (usable) IFML document.

```
<Ifml>
<Head/>
<Body/>
</Ifml>
```

*Note that <tag/> = <tag></tag>

All IFML files must be well-formed, and this is achieved by making sure all tags are closed. A non-well-formed IFML document is provided as an example below. It should be noted that the "Head" and "Body" tags are not closed.

```
<Ifml>
<Head>
<Body>
</Ifml>
```

A well-formed IFML document is provided as an example below.

```
<Ifml>
<Head>
<Title>This is a well-formed IFML page</Title>
</Head>
<Body>
</Body>
</Ifml>
```

Tags are case sensitive, therefore the case of an opening tag must match the case of a closing tag. Also, all tags can be written in any combination of upper or lower case. All attributes of all tags must be in lower case and the values assigned to attributes must be in double quotes.

The various tags and attributes will now be described, with examples to illustrate use where appropriate.

Body Tag <Body> Ifml – Body

The Body Tag is the main tag where all the contents of the IFML document is expressed. Under the Body Tag, the attributes allowed are provided in TABLE 2 below.

Body	Bgcolor	Optional
	Left	Optional
	Top	Optional

TABLE 2

- Ifml - Body – **Bgcolor** - attribute

The Bgcolor attribute of the Body tag refers to the background color of the web page produced by the IFML document. All background color will reflect according to the body background color attribute, when not overloaded by a local background color attribute of some other tags. Acceptable values for the Bgcolor attribute are detailed in TABLE 3 below.

Bgcolor	Aqua	Olive
	Black	Purple
	Blue	Red
	Fuchsia	Silver
	Gray	Teal
	Green	White
	Lime	Yellow
	Maroon	#000000 - #FFFFFF
	Navy	

TABLE 3

The Bgcolor attribute can either be set by entering the English name of the color desired, or the hexadecimal value of the color. The # sign is to denote the attribute value as a hexadecimal. The hexadecimal value is then represented by 6 digits, with the red, green, blue values

taking 2 digits respectively. Black in hexadecimal is #00000, while White is #FFFFFF. An example is provided below.

```
<Body bgcolor="Aqua">
</Body>
```

Another example follows.

```
<Body bgcolor="#012345">
</Body>
```

- Ifml - Body – **left** - attribute

The left attribute of the Body tag refers to the left margin of the IFML document, and this attribute is only relevant when the IFML document is transcoded into HTML documents. This attribute will have no effect during the transcoding of WML documents. It should be noted that the minimum value of the left attribute is 0 and there is no maximum value limit. Acceptable values for this attribute are provided in TABLE 4, which follows.

left	0 – Infinity

TABLE 4

An example is provided below.

```
<Body left="0">
</Body>
```

Another example follows.

```
<Body left="1000">
</Body>
```

- Ifml - Body – **top** - attribute

The top attribute of the Body tag refers to the top margin of the IFML document, and this attribute is only relevant when the IFML document is transcoded into HTML documents. Again, this attribute will have no effect during the transcoding of WML documents. It should be noted that the minimum value of the top attribute is 0 and there is no maximum value limit. Acceptable values for this attribute are provided in TABLE 5, which follows.

top	0 – Infinity

TABLE 5

An example is provided below.

```
<Body top="0">
</Body>
```

Another example follows.

```
<Body top="1000">
</Body>
```

Column Tag <Column> Ifml – Table – Row - Column

The Column is a child tag of the Row tag, which in turn is a child tag of the Table tag. The Column tag is the content area of the Table tag. Under the Column tag, the allowed attributes are detailed in TABLE 6 below.

Column	align	Optional
	bgcolor	Optional
	colspan	Optional
	height	Optional
	rowspan	Optional
	valign	Optional
	width	Optional

TABLE 6

- Ifml – Table – Row - Column – **align**- attribute

The align attribute of the Column tag refers to the horizontal alignment of the content within the content area. If the align attribute is not overloaded, the alignment of the content will be left by default. Acceptable values are provided in TABLE 7 below.

Align	Left Center Right

TABLE 7

An example is provided below.

```
<Column align="right">
</Column>
```

- Ifml – Table – Row – Column – **bgcolor** - attribute

The bgcolor attribute of the Column tag refers to the background color of the content area, and will overload the bgcolor attribute from the Body Tag. The bgcolor attribute can either be set by entering the English name of the color desired or the hexadecimal value of the color. The # sign is to denote the attribute value as a hexadecimal. The hexadecimal value is then represented by 6 digits, with the red, green, blue values taking 2 digits respectively. Black in hexadecimal is #00000, while White is #FFFFFF. Acceptable values are detailed in TABLE 8 below.

Bgcolor	Aqua	Olive
	Black	Purple
	Blue	Red
	Fuchsia	Silver
	Gray	Teal
	Green	White
	Lime	Yellow
	Maroon	#000000 - #FFFFFF
	Navy	

TABLE 8

An example follows.

```
<Column bgcolor="Aqua">
</Column>
```

Another example follows.

```
<Column bgcolor="#012345">
</Column>
```

- Ifml – Body – Table – Row – Column – **colspan** - attribute

The colspan attribute of the Column tag refers to column spanning across multiple columns, and acceptable values are provided in TABLE 9 below.

colspan	1- Total Number of Columns

TABLE 9

An example is provided below.

```
<Column colspan="2">
</Column>
```

- Ifml – Body – Table – Row – Column – **height** - attribute

The height attribute of the Column tag refers to the vertical dimensions of the content area. The minimum value of the height attribute is 0 and there is no maximum value limit. As for the value in percentage, 100% will mean the maximum permissible value, while 0% the minimum. Acceptable values are provided in TABLE 10 below.

height	0 - Infinity 0% - 100%

TABLE 10

An example follows.

```
<Column height="3">
</Column>
```

Another example follows.

```
<Column height="43%">
</Column>
```

- Ifml – Body – Table – Row – Column – **rowspan** - attribute

The rowspan attribute of the Column tag refers to the row spanning across multiple rows, and acceptable values are detailed in TABLE 11 below.

rowspan	1- Total Number of Rows

TABLE 11

An example is provided below.

```
<Column rowspan="2">
</Column>
```

- Ifml – Body – Table – Row – Column – **valign** - attribute

The valign attribute of the Column tag refers to the vertical alignment of the content within the content area. If the valign attribute is not overloaded, the alignment of the content will be middle by default. Acceptable values are provided in TABLE 12 below.

valign	top middle bottom baseline

TABLE 12

An example follows.

```
<Column valign="top">
</Column>
```

- Ifml – Body – Table – Row – Column – **width** - attribute

The width attribute of the Column tag refers to the horizontal dimensions of the content area. The minimum value of the height attribute is 0 and there is no maximum value limit. As for the value in percentage, 100% will mean the maximum permissible value, while 0% the minimum. Acceptable values are provided in TABLE 13 below.

width	0 - Infinity 0% - 100%

TABLE 13

An example is provided below.

```
<Column width="3">
</Column>
```

Another example follows.

```
<Column width="43%">
</Column>
```

Form Tag <Form> Ifml – Body – Form

The Form tag is the tag where content developers denote data posting. Display-components such as textboxes or buttons will fall within the Form tag. The attributes under the form tag are detailed in TABLE 14 below.

Form	action method name	Compulsory Compulsory Optional

TABLE 14

- Ifml – Body – Form – **action** - attribute

The action attribute of the Form tag refers to the file or location to which data that is submitted is sent. A URL or a filename for the action file can be specified as provided in TABLE 15 below.

action	URL

TABLE 15

An example follows.

```
<Form action=http://www.microsoft.com/action.jsp>
</Form>
```

Another example follows.

```
<Form action="action.jsp">
</Form>
```

- Ifml – Body – Form – **method** - attribute

The method attribute of the Form tag refers to the way that data or content within the Form tag can be sent to the action address. When using "get", data gets appended to the end of the URL. Acceptable values are detailed in TABLE 16 below.

method	get
	post

TABLE 16

An example is provided below.

```
<Form method="post">
</Form>
```

Another example is provided below.

```
<Form method="get">
</Form>
```

- Ifml – Body – Form – **name** - attribute

The name attribute of the Form tag refers to the name that is given to the Form tag. Names can be any length and any word that doesn't start with a number or special character, and acceptable values are provided in TABLE 17 below.

name	*

TABLE 17

An example follows.

```
<Form name="form1">
</Form>
```

Group Tag <Group> Ifml – Body – Form - Group

The Group tag is a child of the Form tag, and is used in conjunction with radio buttons for WML documentation transcoding purposes. The attribute under the Group tag are detailed in TABLE 18 below.

Group	name	Optional

TABLE 18

- Ifml – Body – Form – Group – **name** - attribute

The name attribute of the Group tag allows for the naming of the Group tag. Names can be any length and any word that doesn't start with a number or a special character, and an example is provided below.

```
<Group name="radio1">
<Input type="radio" name="radio1">1</option>
<Input type="radio" name="radio1">2</option>
<Input type="radio" name="radio1">3</option>
<Input type="radio" name="radio1">4</option>
</Group>
```

Head Tag <Head> Ifml – Head

The Head tag is a child of the Title tag, and the Head tag together with the Title tag provide the name of the IFML document. There are no attributes under the Head Tag, and an example is provided below.

```
<Head>
    <Title>Ifml Rocks</Title>
</Head>
```

IFML Tag <Ifml> Ifml

The Ifml tag is the root element of the Ifml document object model. All Ifml files must be denoted at the start and at the end with the Ifml tag. There are no attributes under the Ifml Tag, and an example is provided below.

```
<Ifml>
  .
  .
  .
</Ifml>
```

Image Tag <Image> Ifml – Body - Image

The Image tag is used for inserting graphical images into the Ifml document. The attributes of the Image tag are detailed in TABLE 19 below.

Image	alt	Compulsory
	border	Optional
	height	Optional
	hspace	Optional
	src	Compulsory
	wapsrc	Optional
	width	Optional

TABLE 19

- Ifml – Body – Image – **alt** - attribute

The alt attribute refers to the alternative text for the image insert should the image not load or when the mouse is moved over the image. It provides a caption. The length and nature of the alternative text are not constrained, and an example is provided below.

<Image src="s1.gif" alt="Super One"/>

- Ifml –Body - Image – **border** - attribute

The border attribute of the Image tag refers to the border around an image if it becomes a child of the Link tag. When this attribute is set to 0, there will be no border available, whilst when set to 1, a border is made available. Acceptable values are detailed in TABLE 20 below.

border	1
	0

TABLE 20

An example follows.

<Image src="s1.gif" alt="" border="0"/>

- Ifml – Body – Image – **Height** - attribute

The height attribute of the Image tag refers to the vertical dimension of the image. The minimum value of the height attribute is 0 and there

is no maximum value limit. As for the value in percentage, 100% will sets the maximum permissible value, while 0% the minimum. Acceptable values are detailed in TABLE 21 below.

height	0 - Infinity 0% - 100%

TABLE 21

An example follows.

```
<Image src="s1.gif" alt="" height="80%"/>
```

Another example follows.

```
<Image src="s1.gif" alt="" height="123"/>
```

- Ifml – Body – Image – **hspace** - attribute

The hspace attribute of the Image tag refers to the horizontal spacing of the image in the Ifml document, and acceptable values are detailed in TABLE 22 below.

hspace	0 - 12

TABLE 22

An example is provided below.

```
<Image src="s1.gif" alt=""/>
```

- Ifml – Body – Image – **src** - attribute

The src attribute of the Image tag refers to the location or source of the image file of the image that is to be inserted into the Ifml document. In one embodiment only GIFS and JPEG image files can be inserted into an Ifml document, hence the acceptable values detailed in TABLE 23 below.

src	*.gif *.jpg

TABLE 23

An example follows.

```
<Image src="s1.gf" alt=""/>
```

- Ifml – Body – Image – **wapsrc** - attribute

The wapsrc attribute of the Image tag refers to the location of the wbmp file. This is used for WML transcoding purposes. In one

embodiment the wapsrc attribute can only accommodate wbmp files, hence the example below.

<Image src="s1.gif" alt="" wapsrc="s1.wbmp"/>

- Ifml – Body – Image – **width** - attribute

The width attribute of the Image tag refers to the horizontal dimension of the image. The minimum value of the width attribute is 0 and there is no maximum value limit. As for the value in percentage, 100% will mean the maximum permissible value, while 0% the minimum. Acceptable values are detailed in TABLE 24 below.

width	0 - Infinity 0% - 100%

TABLE 24

An example is provided below.

<Image src="s1.gif" alt="" width="80%"/>

Another example follows.

<Image src="s1.gif" alt="" width="123"/>

Input Tag <Input> Ifml – Body – Form – Input

The Input type is a child of the Form tag. The Input tag facilitates use of buttons, checkboxes and text fields for data submission purposes. The attributes are detailed in TABLE 25 below.

Input	Name	Compulsory
	Type	Compulsory
	Value	Compulsory

TABLE 25

- Ifml – Body – Form – Input – **name** - attribute

The name attribute of the Input tag refers to the name of the Input tag. The name attribute allows data to be identified when submitted for processing. Names can be any length and any word that doesn't start with a number or special characters, and an example is provided below.

<input type="text" value="" name="text1"/>

- Ifml – Body – Form – Input – **type** - attribute

The type attribute of the Input tag refers to the kind of input components that an author wishes to use to solicit data. Acceptable values are shown in TABLE 26 below.

type	checkbox
	password
	radio
	submit
	Text

TABLE 26

A first example follows.

```
<Input type="checkbox" value="aan" name="checker"/>
```

A second example follows.

```
<Input type="password" value="aan" name="pass"/>
```

A third example follows.

```
<Group name="checker">
<Input type="radio" value="aan1" name="rchecker">1</input>
<Input type="radio" value="aan2" name="rchecker">2</input>
<Input type="radio" value="aan3" name="rchecker">3</input>
</Group>
```

A fourth example follows.

```
<Input type="submit" value="aan" name="sub"/>
```

A fifth example follows.

```
<Input type="text" value="aan" name="texter"/>
```

- Ifml – Body – Form – Input – **value** - attribute

The value attribute of the Input tag refers to the value stored in the Input tag, and an example is provided below.

```
<input type="text" value="abc" name="text1"/>
```

JSP Tag <Jsp> Ifml – Jsp

The Jsp tag is used for embedding Jsp codes into the IFML document. Jsp tags can be put in any part of the IFML document, and there are no attributes for the Jsp Tag. An example follows.

```
<Jsp>
 for (int i=10;i>0;i--)
{
        </Jsp>
        <Jsp>=i</Jsp>
        <Jsp>
}
</Jsp>
```

LineBreak Tag <LineBreak> Ifml – Body – LineBreak

The LineBreak tag is used to induce a carriage return in the IFML document. The attributes for the LineBreak tag are not provided here, however, an example follows.

```
<LinkBreak/>
```

Link Tag <Link> Ifml – Body – Link

The Link tag is used for creating a hyperlink between an IFML document to any other document types. There is one attribute for The Link tag, and it is detailed in TABLE 27 below.

Link	href	Compulsory

TABLE 27

- Ifml – Body – Link – **href** - attribute

The href attribute of the Link tag refers to the target url that the link leads to and acceptable value is provided in TABLE 28.

href	URL

TABLE 28

An example is provided below.

```
<Link href="http://www.google.com">Click to Proceed</Link>
```

NBSP Tag <NBSP> Ifml – Body – NBSP

The NBSP Tag is a formatting tag known as the Non-Breaking Space Tag. This tag will be transcoded for WML documents. There are no attributes for the NBSP Tag, and an example follows.

```
<NBSP/>
```

Option Tag <Row> Ifml – Form – Select – Option

The Option tag is a child of the Select tag. When used together, the Select and Option tags generate either a drop-down list or list box. There is one attribute for the Option tag, as detailed in TABLE 29 below.

Option	value	Compulsory

TABLE 29

- Ifml – Form – Select – Option – **value** - attribute

The value attribute is the equivalent of the input-value attribute. It is used to hold the value of the data component selected, and an example is provided below.

```
<Select name="choice">
<Option value="1">One</Option>
<Option value="2">Two</Option>
</Select>
```

Row Tag <Row> Ifml – Table – Row

The Row tag is the parent of the Column tag and child of the Table tag. Under the Row tag, the attributes that are provided in TABLE 30 are available.

Row	Align	Optional
	bgcolor	Optional
	colspan	Optional
	height	Optional
	rowspan	Optional
	Valign	Optional
	Width	Optional

TABLE 30

- Ifml – Table – Row – **align** - attribute

The align attribute of the Row tag refers to the horizontal alignment of the content within the content area. If the align attribute is not overloaded, the alignment of the content will be left by default. Acceptable values are provided in TABLE 31 below.

Align	Left Center Right

TABLE 31

An example follows.

```
<Row align="right">
</Row>
```

- Ifml – Table – Row – **bgcolor** - attribute

The bgcolor attribute of the Row tag refers to the background color of the content area. This will overload the bgcolor attribute from the Body tag. The bgcolor attribute can either be set by entering the English name of the color desired or the hexadecimal value of the color. The # sign is to denote the attribute value as a hexadecimal. The hexadecimal value is then represented by 6 digits, with the red, green, blue values taking 2 digits respectively. Black in hexadecimal is #00000, while White is #FFFFFF. Acceptable values are detailed in TABLE 32 below.

Bgcolor	Aqua	Olive
	Black	Purple
	Blue	Red
	Fuchsia	Silver
	Gray	Teal
	Green	White
	Lime	Yellow
	Maroon	#000000 - #FFFFFF
	Navy	

TABLE 32

An example follows.

```
<Row bgcolor="Aqua">
</Row>
```

Another example follows.

```
<Row bgcolor="#012345">
</Row>
```

- Ifml – Body – Table – Row – **colspan** - attribute

The colspan attribute of the Row tag refers to a column spanning across multiple columns, and acceptable values are provided in TABLE 33 below.

colspan	1- Total Number of Columns

TABLE 33

An example is provided below.

```
<Row colspan="2">
</Row>
```

- Ifml – Body – Table – Row – **height** - attribute

The height attribute of the Row tag refers to the vertical dimensions of the content area. The minimum value of the height attribute is 0 and there is no maximum value limit. In percentage, 100% will mean the maximum permissible value, while 0% the minimum. Acceptable values are provided in TABLE 34 below.

height	0 - Infinity 0% - 100%

TABLE 34

A first example follows.

```
<Row height="3">
</Row>
```

A second example follows.

```
<Row height="43%">
</Row>
```

- Ifml – Body – Table – Row – **rowspan** - attribute

The rowspan attribute of the Row tag refers to a row spanning across multiple rows, and acceptable values are detailed in TABLE 35 below.

rowspan	1- Total Number of Rows

TABLE 35

An example follows.

```
<Row rowspan="2">
</Row>
```

- Ifml – Body – Table – Row – **valign** - attribute

The valign attribute of the Row tag refers to the vertical alignment of the content within the content area. If the valign attribute is not overloaded, the alignment of the content will be in the middle of the

content area by default. Acceptable values are provided in TABLE 36 below.

valign	top middle bottom baseline

TABLE 36

An example is provide below.

```
<Row valign="top">
</Row>
```

- Ifml – Body – Table – Row – **width** - attribute

The width attribute of the Row tag refers to the horizontal dimensions of the content area. Please note that the minimum value of the height attribute is 0 and there is no maximum value limit. As for the value in percentage, 100% will mean the maximum permissible value, while 0% the minimum. Acceptable values are detailed in TABLE 37 below.

Width	0 - Infinity 0% - 100%

TABLE 37

A first example follows.

```
<Row width="3">
</Row>
```

A second example follows.

```
<Row width="43%">
</Row>
```

Select Tag <Row> Ifml – Form – Select

The Select tag is a parent tag of the Option tag. Together the Select tag and the Option tag will be generated into either a dropdown list or a list box. The Select tag has one attribute as provided in TABLE 38 below.

Option	value	Compul sory

TABLE 38

- Ifml – Form – Select – Option – **value** - attribute

The value attribute is the equivalent of the input-value attribute. It is used to hold the value of the data component selected, and an example is provided below.

```
<Select name="choice">
<Option value="1">One</Option>
<Option value="2">Two</Option>
</Select>
```

Title Tag <Title> Ifml – Head – Title

The Title tag is a tag used to denote the title of the IFML document, this tag is a child of the Head tag and has no attributes. An example is provided below.

```
<Head>
<Title>Welcome to PDA.com</Title>
</Head>
```

WapBreak Tag <WapBreak> Ifml – Body - WapBreak

The WapBreak tag is a formatting tag used to induce a carriage return. The fundamental difference between the LineBreak tag and WapBreak tag, is that the WapBreak only induces carriage returns in WML documents. There is no attributes for this tag and an example is provided below.

```
<Wapbreak/>
```

An example using an IFML document will now be provided, where the contents of the IFML file is listed. Following that the transcoded HTML file contents and the transcoded WML file contents are listed, and screen shots of the resultant web pages using the HTML and WML file are provided.

The following is the listing of an IFML file that may be produced using the authoring tool 100, after an author has designed a web page therewith.

IFML FILE

```
<?xml version="1.0" encoding="utf-8"?>
<ifml>

   <head><title>Welcome to PDA.com</title></head>
   <body bgcolor="#ffffff" topmargin="0" leftmargin="0">

   <table cellspacing="0" cellpadding="0" width="100%" height="100%">
      <row height="80" bgcolor="#7070cc"><column span="2">
         <table cellpadding="4">
            <row><column align="left">
                 <image alt="banner" src="title2.gif" wapsrc="title1.wbmp"/>
                 <wapbreak/>
                 </column>
            </row>
            <row height="10"><column><font face="Verdana" size="2">
               <link href="home.ifml">Home</link> |
                  <link href="about.ifml">About Us</link> |
                  <link href="highlights.ifml">Weekly Highlights</link> |
                  <link href="forums.ifml">Forums</link> |
                  <link href="poll.ifml">Poll</link> |
                  </font></column><wapbreak/>
            </row>
         </table></column></row>

      <row>
        <column bgcolor="#f1f1f1" valign="top" width="150">
          <table cellspacing="0" cellpadding="4">
            <row>
               <column>
                  <font face="Verdana" size="2">
                  <b>
                     <img hspace="5" src="arrow2.gif" wapsrc="arrow2.wbmp"/>
                     Search
                  </b>
                  </font>
               </column>
            </row><wapbreak/>
            <row>
               <column>
                  <font face="Verdana" size="2">
                     <image alt="" hspace="10" src="arrow3.gif" wapsrc="arrow2.wbmp"/>
                     <link href="http://www.palm.com">Palm</link>
                  </font><wapbreak/>
               </column>
            </row>
            <row>
               <column>
                  <font face="Verdana" size="2">
                     <image alt="" hspace="10" src="arrow3.gif" wapsrc="arrow2.wbmp"/>
                     <link href="http://www.sonystyle.com/micros/clie">Sony</link>
                  </font><wapbreak/>
               </column>
            </row>
            <row>
               <column>
                  <font face="Verdana" size="2">
                     <image alt="" hspace="10" src="arrow3.gif" wapsrc="arrow2.wbmp"/>
                     <link href="http://www.cas-pda.com">Casio</link>
                  </font><wapbreak/>
               </column>
            </row>
            <row>
               <column>
                  <font face="Verdana" size="2">
                     <image alt="" hspace="10" src="arrow3.gif" wapsrc="arrow2.wbmp"/>
                     <link
href="http://www.compaq.com/products/handhelds/pocketpc">Compaq</link>
                  </font><wapbreak/>
               </column>
               </row>
               <row>
```

```
                    <column>
                        <font face="Verdana" size="2">
                            <image alt="" hspace="10" src="arrow3.gif"
wapsrc="arrow2.wbmp"/>
                            <link href="http://www.handspring.com">Handspring</link>
                        </font><wapbreak/>
                    </column>
                </row>
                <row>
                    <column>
                        <font face="Verdana" size="2">
                            <image alt="" hspace="10" src="arrow3.gif"
wapsrc="arrow2.wbmp"/>
                            <link href="http://www.hp.com/jornada">Hewlett Packard</link>
                        </font><wapbreak/>
                    </column>
                </row><wapbreak/>
            </table>
            <table cellspacing="0" cellpadding="4">
                <row>
                    <column>
                        <font face="Verdana" size="2">
                            <b>
                            <img hspace="5" src="arrow2.gif" wapsrc="arrow2.wbmp"/>
                            Related Links
                            </b>
                        </font>
                    </column>
                </row><wapbreak/>
                <row>
                    <column>
                        <font face="Verdana" size="2">
                            <image alt="" hspace="10" src="arrow3.gif"
wapsrc="arrow2.wbmp"/>
                            <link href="http://www.pdabuzz.com">PDABuzz.com</link>
                        </font><wapbreak/>
                    </column>
                </row>
                <row>
                    <column>
                        <font face="Verdana" size="2">
                            <image alt="" hspace="10" src="arrow3.gif"
wapsrc="arrow2.wbmp"/>
                            <link href="http://www.pdastreet.com">PDA Street</link>
                        </font><wapbreak/>
                    </column>
                </row>
                <row>
                    <column>
                        <font face="Verdana" size="2">
                            <image alt="" hspace="10" src="arrow3.gif"
wapsrc="arrow2.wbmp"/>
                            <link
href="http://www.geek.com/pdageek/pdamain.htm">Geek.com</link>
                        </font><wapbreak/>
                    </column>
                </row>
                <row>
                    <column>
                        <font face="Verdana" size="2">
                            <image alt="" hspace="10" src="arrow3.gif"
wapsrc="arrow2.wbmp"/>
                            <link href="http://www.pdabuyersguide.com/">PDA Buyers
Guide</link>
                        </font><wapbreak/>
                    </column>
                </row>
                <row>
                    <column>
                        <font face="Verdana" size="2">
                            <image alt="" hspace="10" src="arrow3.gif"
wapsrc="arrow2.wbmp"/>
                            <link href="http://www.pdaforums.com/">PDA Forums.com</link>
                        </font><wapbreak/>
```

```
            </column>
          </row><wapbreak/>
        </table>
      </column>
      <column valign="top">
        <table width="60%" cellpadding="4">
          <row>
            <column><wapbreak/>
              <image hspace="2" src="arrow2.gif" wapsrc="arrow2.wbmp" alt=""/>
            </column>
            <column>
              <font face="verdana" size="2"><b>Two hot wireless PDAs</b></font>
            </column>
            <column><wapbreak/>
              <font face="verdana" size="2">8/2/2002</font><wapbreak/>
            </column>
            <column rowspan="2">
              <img src="headline1.jpg" alt="headlines"/>
            </column>
          </row>
          <row>
            <column span="3">
              <font face="verdana" size="2">
              Handspring's Treo, which melds a cell phone and a PDA,
              and Palm's new, wireless i705 made headlines. But are
              they as good as their hype?
              </font>
            </column>
          </row>
          <row bgcolor="black">
            <column span="4"/>
          </row>
          <row>
            <column><wapbreak/>
              <image hspace="2" src="arrow2.gif" wapsrc="arrow2.wbmp" alt=""/>
            </column>
            <column>
              <font face="verdana" size="2">
                <b>Handspring Treo to offer improved corporate solutions</b>
              </font>
            </column>
            <column><wapbreak/>
              <font face="verdana" size="2">5/2/2002</font><wapbreak/>
            </column>
            <column rowspan="2">
              <img src="headline2.gif" alt="headlines"/>
            </column>
          </row>
          <row>
            <column span="3">
              <font face="verdana" size="2">
              According to CRM Daily, Handspring is pushing its new
              Treo PDA Phone into the corporate world. Working with
              partners like Aether Systems, AvantGo and Wireless Knowledge
              in an effort to improve the solutions offered for mobile
              sales force automation, customer relation management and
              mobile data access. According to the article, "Handspring
              said its goal is to make Treo and Visor handhelds standard
              mobile business platforms."
              </font>
            </column>
          </row>
          <row bgcolor="black">
            <column span="4"/>
          </row>
          <row>
            <column><wapbreak/>
              <image hspace="2" src="arrow2.gif" wapsrc="arrow2.wbmp" alt=""/>
            </column>
            <column>
              <font face="verdana" size="2">
                <b>Sony unveils new handheld - CLIE T415</b>
              </font>
            </column>
```

```
                    <column><wapbreak/>
                        <font face="verdana" size="2">2/2/2002</font><wapbreak/>
                    </column>
                    <column rowspan="2">
                        <img src="headline3.gif" alt="headlines"/>
                    </column>
                </row>
                <row>
                    <column span="3">
                        <font face="verdana" size="2">
                        Sony have unveiled a new model of their Palm OS based PDA,
                        the CLIE T415. The new handheld will incorporate standard
                        Sony PDA features including their Jog Shuttle, and memory
                        stick support. In addition, the CLIE now ships with new
                        software that allows it to double as a remote control for
                        your household audio/video equipment.
                        </font>
                    </column>
                </row>
                <row bgcolor="black">
                    <column span="4"/>
                </row>
            </table>
        </column>
        </row>
        <row bgcolor="#7070cc">
        <column height="25" span="2">
            <table bgcolor="#7070cc" cellpadding="4">
                <row>
                    <column>
                        <font face="Verdana" size="2">
                        Copyright 2002 PDA.com Inc. All rights reserved.
                        </font>
                    </column>
                </row>
            </table>
        </column>
        </row>
    </table>

</body>
</ifml>
```

The IFML file is then transcoded by the transcoding system 400, where for illustration purposes the IFML processor 425 has determined that the requesting device requires that the requested web page be in HTML. The resulting HTML file listing is provided below.

HTML FILE TRANSCODED FROM THE IFML FILE

```
<HTML>

<HEAD><TITLE>Welcome to PDA.com</TITLE></HEAD>
<BODY BGCOLOR="#ffffff" TOPMARGIN="0" LEFTMARGIN="0">

<TABLE CELLSPACING="0" CELLPADDING="0" WIDTH="100%" HEIGHT="100%">
   <TR HEIGHT="80" BGCOLOR="#7070cc">
      <TD COLSPAN="2">
         <TABLE CELLPADDING="4">
            <TR>
               <TD ALIGN="left">
                  <IMG ALT="banner" SRC="title2.gif" WAPSRC="title1.wbmp">
               </TD>
            </TR>
            <TR HEIGHT="10">
```

```
                <TD>
                    <FONT FACE="Verdana" SIZE="2">
                    <A HREF="../../UniTranscode.jsp?file=home.ifml">Home</A> |
                    <A HREF="../../UniTranscode.jsp?file=about.ifml">About Us</A> |
                    <A HREF="../../UniTranscode.jsp?file=highlights.ifml">Weekly
Highlights</A> |
                    <A HREF="../../UniTranscode.jsp?file=forums.ifml">Forums</A> |
                    <A HREF="../../UniTranscode.jsp?file=poll.ifml">Poll</A> |
                    </FONT>
                </TD>
            </TR>
        </TABLE>
    </TD>
</TR>
<TR>
    <TD BGCOLOR="#f1f1f1" VALIGN="top" WIDTH="150">
        <TABLE CELLSPACING="0" CELLPADDING="4">
            <TR>
                <TD>
                    <FONT FACE="Verdana" SIZE="2">
                    <B><IMG HSPACE="5" SRC="arrow2.gif"
WAPSRC="arrow2.wbmp"></IMG>Search</B>
                    </FONT>
                </TD>
            </TR>
            <TR>
                <TD>
                    <FONT FACE="Verdana" SIZE="2">
                        <IMG ALT="" HSPACE="10" SRC="arrow3.gif" WAPSRC="arrow2.wbmp">
                        <A HREF="http://www.palm.com">Palm</A>
                    </FONT>
                </TD>
            </TR>
            <TR>
                <TD>
                    <FONT FACE="Verdana" SIZE="2">
                        <IMG ALT="" HSPACE="10" SRC="arrow3.gif" WAPSRC="arrow2.wbmp">
                        <A HREF="http://www.sonystyle.com/micros/clie">Sony</A>
                    </FONT>
                </TD>
            </TR>
            <TR>
                <TD>
                    <FONT FACE="Verdana" SIZE="2">
                    <IMG ALT="" HSPACE="10" SRC="arrow3.gif" WAPSRC="arrow2.wbmp">
                    <A HREF="http://www.cas-pda.com">Casio</A>
                    </FONT>
                </TD>
            </TR>
            <TR>
                <TD>
                    <FONT FACE="Verdana" SIZE="2">
                    <IMG ALT="" HSPACE="10" SRC="arrow3.gif" WAPSRC="arrow2.wbmp">
                    <A HREF="http://www.compaq.com/products/handhelds/pocketpc">Compaq</A>
                    </FONT>
                </TD>
            </TR>
            <TR>
                <TD>
                    <FONT FACE="Verdana" SIZE="2">
                    <IMG ALT="" HSPACE="10" SRC="arrow3.gif" WAPSRC="arrow2.wbmp">
                    <A HREF="http://www.handspring.com">Handspring</A>
                    </FONT>
                </TD>
            </TR>
            <TR>
                <TD>
                    <FONT FACE="Verdana" SIZE="2">
                    <IMG ALT="" HSPACE="10" SRC="arrow3.gif" WAPSRC="arrow2.wbmp">
                    <A HREF="http://www.hp.com/jornada">Hewlett Packard</A>
                    </FONT>
                </TD>
            </TR>
        </TABLE>
```

```
<TABLE CELLSPACING="0" CELLPADDING="4">
   <TR>
      <TD>
         <FONT FACE="Verdana" SIZE="2">
         <B>
         <IMG HSPACE="5" SRC="arrow2.gif" WAPSRC="arrow2.wbmp"></IMG>
         Related Links
         </B>
         </FONT>
      </TD>
   </TR>
   <TR>
      <TD>
         <FONT FACE="Verdana" SIZE="2">
         <IMG ALT="" HSPACE="10" SRC="arrow3.gif" WAPSRC="arrow2.wbmp">
         <A HREF="http://www.pdabuzz.com">PDABuzz.com</A>
         </FONT>
      </TD>
   </TR>
   <TR>
      <TD>
         <FONT FACE="Verdana" SIZE="2">
         <IMG ALT="" HSPACE="10" SRC="arrow3.gif" WAPSRC="arrow2.wbmp">
         <A HREF="http://www.pdastreet.com">PDA Street</A>
         </FONT>
      </TD>
   </TR>
   <TR>
      <TD>
         <FONT FACE="Verdana" SIZE="2">
         <IMG ALT="" HSPACE="10" SRC="arrow3.gif" WAPSRC="arrow2.wbmp">
         <A HREF="http://www.geek.com/pdageek/pdamain.htm">Geek.com</A>
         </FONT>
      </TD>
   </TR>
   <TR>
      <TD>
         <FONT FACE="Verdana" SIZE="2">
         <IMG ALT="" HSPACE="10" SRC="arrow3.gif" WAPSRC="arrow2.wbmp">
         <A HREF="http://www.pdabuyersguide.com/">PDA Buyers Guide</A>
         </FONT>
      </TD>
   </TR>
   <TR>
      <TD>
         <FONT FACE="Verdana" SIZE="2">
         <IMG ALT="" HSPACE="10" SRC="arrow3.gif" WAPSRC="arrow2.wbmp">
         <A HREF="http://www.pdaforums.com/">PDA Forums.com</A>
         </FONT>
      </TD>
   </TR>
</TABLE>
</TD>
<TD VALIGN="top">
   <TABLE WIDTH="60%" CELLPADDING="4">
      <TR>
         <TD>
            <IMG HSPACE="2" SRC="arrow2.gif" WAPSRC="arrow2.wbmp" ALT="">
         </TD>
         <TD>
            <FONT FACE="verdana" SIZE="2">
            <B>Two hot wireless PDAs</B>
            </FONT>
         </TD>
         <TD>
            <FONT FACE="verdana" SIZE="2">8/2/2002</FONT>
         </TD>
         <TD ROWSPAN="2">
            <IMG SRC="headline1.jpg" ALT="headlines"></IMG>
         </TD>
      </TR>
      <TR>
         <TD COLSPAN="3">
            <FONT FACE="verdana" SIZE="2">
```

```
                    Handspring's Treo, which melds a cell phone and a PDA, and Palm's new,
wireless i705
                    made headlines. But are they as good as their hype?
                    </FONT>
                 </TD>
              </TR>
              <TR BGCOLOR="black">
                 <TD COLSPAN="4"></TD>
              </TR>
              <TR>
                 <TD>
                    <IMG HSPACE="2" SRC="arrow2.gif" WAPSRC="arrow2.wbmp" ALT="">
                 </TD>
                 <TD>
                    <FONT FACE="verdana" SIZE="2">
                    <B>Handspring Treo to offer improved corporate solutions</B>
                    </FONT>
                 </TD>
                 <TD>
                    <FONT FACE="verdana" SIZE="2">5/2/2002</FONT>
                 </TD>
                 <TD ROWSPAN="2">
                    <IMG SRC="headline2.gif" ALT="headlines"></IMG>
                 </TD>
              </TR>
              <TR>
                 <TD COLSPAN="3">
                    <FONT FACE="verdana" SIZE="2">
                    According to CRM Daily, Handspring is pushing its new Treo PDA Phone
into the corporate
                    world. Working with partners like Aether Systems, AvantGo and Wireless
Knowledge in an
                    effort to improve the solutions offered for mobile sales force
automation, customer
                    relation management and mobile data access. According to the article,
"Handspring
                    said its goal is to make Treo and Visor handhelds standard mobile
business platforms."
                    </FONT>
                 </TD>
              </TR>
              <TR BGCOLOR="black">
                 <TD COLSPAN="4"></TD>
              </TR>
              <TR>
                 <TD>
                    <IMG HSPACE="2" SRC="arrow2.gif" WAPSRC="arrow2.wbmp" ALT="">
                 </TD>
                 <TD>
                    <FONT FACE="verdana" SIZE="2">
                    <B>Sony unveils new handheld - CLIE T415</B>
                    </FONT>
                 </TD>
                 <TD>
                    <FONT FACE="verdana" SIZE="2">2/2/2002</FONT>
                 </TD>
                 <TD ROWSPAN="2">
                    <IMG SRC="headline3.gif" ALT="headlines"></IMG>
                 </TD>
              </TR>
              <TR>
                 <TD COLSPAN="3">
                    <FONT FACE="verdana" SIZE="2">
                    Sony have unveiled a new model of their Palm OS based PDA, the CLIE
T415. The new
                    handheld will incorporate standard Sony PDA features including their
Jog Shuttle, and
                    memory stick support. In addition, the CLIE now ships with new
    software that allows
                    it to double as a remote control for your household audio/video
equipment.
                    </FONT>
                 </TD>
              </TR>
```

```
            <TR BGCOLOR="black">
               <TD COLSPAN="4"></TD>
            </TR>
         </TABLE>
      </TD>
   </TR>
   <TR BGCOLOR="#7070cc">
      <TD HEIGHT="25" COLSPAN="2">
         <TABLE BGCOLOR="#7070cc" CELLPADDING="4">
            <TR>
               <TD>
                  <FONT FACE="Verdana" SIZE="2">
                  Copyright 2002 PDA.com Inc. All rights reserved.
                  </FONT>
               </TD>
            </TR>
         </TABLE>
      </TD>
   </TR>
</TABLE>

</BODY>
</HTML>
```

FIG. 10A shows a web page produced using the HTML file listed above, as the web page is scrolled on a display of the requesting device.

The IFML file is then transcoded by the transcoding system 400, where for illustration purposes the IFML processor 425 has determined that the requesting device requires that the requested web page be in WML. The resulting WML file listing is provided below.

WML FILE TRANSCODED FROM THE IFML FILE

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN" "http://www.wapforum.org/DTD/wml13.dtd">

<wml>
<card id="Main" title="Welcome to PDA.com" newcontext="true">
<p><img alt="banner" src="title1.wbmp"></img>

<br></br>
<small><a href="../../UniTranscode.jsp?file=home.ifml">Home</a></small> |
<small><a href="../../UniTranscode.jsp?file=about.ifml">About Us</a></small> |
<small><a href="../../UniTranscode.jsp?file=highlights.ifml">Weekly Highlights</a></small> |
<small><a href="../../UniTranscode.jsp?file=forums.ifml">Forums</a></small> |
<small><a href="../../UniTranscode.jsp?file=poll.ifml">Poll</a></small> |

<br></br>Search<br></br>
<img alt="" hspace="10" src="arrow2.wbmp"></img>
  <small><a href="http://www.palm.com">Palm</a></small>
  <br></br>
<img alt="" hspace="10" src="arrow2.wbmp"></img>
  <small><a href="http://www.sonystyle.com/micros/clie">Sony</a></small>
  <br></br>
<img alt="" hspace="10" src="arrow2.wbmp"></img>
  <small><a href="http://www.cas-pda.com">Casio</a></small>
  <br></br>
<img alt="" hspace="10" src="arrow2.wbmp"></img>
  <small><a href="http://www.compaq.com/products/handhelds/pocketpc">Compaq</a></small>
  <br></br>
<img alt="" hspace="10" src="arrow2.wbmp"></img>
  <small><a href="http://www.handspring.com">Handspring</a></small>
  <br></br>
<img alt="" hspace="10" src="arrow2.wbmp"></img>
```

```
<small><a href="http://www.hp.com/jornada">Hewlett Packard</a></small>
<br></br>

<br></br>Related Links<br></br>
<img alt="" hspace="10" src="arrow2.wbmp"></img>
<small><a href="http://www.pdabuzz.com">PDABuzz.com</a></small>
<br></br>
<img alt="" hspace="10" src="arrow2.wbmp"></img>
<small><a href="http://www.pdastreet.com">PDA Street</a></small>
<br></br>
<img alt="" hspace="10" src="arrow2.wbmp"></img>
<small><a href="http://www.geek.com/pdageek/pdamain.htm">Geek.com</a></small>
<br></br>
<img alt="" hspace="10" src="arrow2.wbmp"></img>
<small><a href="http://www.pdabuyersguide.com/">PDA Buyers Guide</a></small>
<br></br>
<img alt="" hspace="10" src="arrow2.wbmp"></img>
<small><a href="http://www.pdaforums.com/">PDA Forums.com</a></small>
<br></br>

<br></br>
<br></br>
<img hspace="2" src="arrow2.wbmp" alt=""></img>
Two hot wireless PDAs
<br></br>8/2/2002<br></br>
Handspring's Treo, which melds a cell phone and a PDA, and Palm's new, wireless i705 made headlines. But are they as good as their
hype?
<br></br>
<img hspace="2" src="arrow2.wbmp" alt=""></img>
Handspring Treo to offer improved corporate solutions
<br></br>5/2/2002<br></br>
According to CRM Daily, Handspring is pushing its new Treo PDA Phone into the corporate world. Working with partners like Aether
Systems, AvantGo and Wireless Knowledge in an effort to improve the solutions offered for mobile sales force automation, customer relation
management and mobile data access. According to the article, "Handspring said its goal is to make Treo and Visor handhelds standard
mobile business platforms."
<br></br>
<img hspace="2" src="arrow2.wbmp" alt=""></img>
Sony unveils new handheld - CLIE T415
<br></br>2/2/2002<br></br>
Sony have unveiled a new model of their Palm OS based PDA, the CLIE T415. The new handheld will incorporate standard Sony PDA
features including their Jog Shuttle, and memory stick support. In addition, the CLIE now ships with new software that allows it to double as a
remote control for your household audio/video equipment.  Copyright 2002 PDA.com Inc. All rights reserved.

</p>
</card>
</wml>
```

FIG. 10B shows a web page produced using the WML file listed above, as the web page is scrolled on a display of the requesting device.

Another example will now be provided to illustrate how user-fillable form components, which are applicable specifically to different markup languages, such as HTML and WML, are transcoded by the IFML processor 425.

IFML FILE WHICH INCLUDES FORM COMPONENTS

```
<ifml>
<head><title>Welcome to Joey's Test Page</title></head>
```

```
<body>
<form method="post" action="form.jsp">
   Username :<input name="textbox" type="text"/><linebreak/>
   Password :<input name="passwordbox" type="password"/><linebreak/>

   Male? :<input type="checkbox" name="Gender" value="True"/><linebreak/>

   Please select your colour. <linebreak/>
     <group name="color">
        <input type="radio" value="1" name="color">Blue</input>
        <input type="radio" value="2" name="color">Red</input>
        <input type="radio" value="3" name="color">Green</input>
     </group>

   <linebreak/> Please choose your number. <linebreak/>
   <select name="Number">
      <option>1</option>
      <option>2</option>
      <option>3</option>
      <option>4</option>
   </select>
   <input type="submit" value="Click To Submit"/>
</form>
</body>
</ifml>
```

HTML FILE TRANSCODED FROM THE IFML FILE WHICH INCLUDES FORM COMPONENTS

```
<HTML>
<HEAD><TITLE>Welcome to Joey's Test Page</TITLE></HEAD>
<BODY>
<FORM METHOD="post" ACTION="form.jsp">
   Username :<INPUT NAME="textbox" TYPE="text"></INPUT><BR>
   Password :<INPUT NAME="passwordbox" TYPE="password"></INPUT><BR>

   Male? :<INPUT TYPE="checkbox" NAME="Gender" VALUE="True"></INPUT><BR>

   Please select your colour. <BR>
   <GROUP NAME="color">
      <INPUT TYPE="radio" VALUE="1" NAME="color">Blue</INPUT>
      <INPUT TYPE="radio" VALUE="2" NAME="color">Red</INPUT>
      <INPUT TYPE="radio" VALUE="3" NAME="color">Green</INPUT>
   </GROUP>
   <BR>
   Please choose your number. <BR>
   <SELECT NAME="Number">
      <OPTION>1</OPTION>
      <OPTION>2</OPTION>
      <OPTION>3</OPTION>
      <OPTION>4</OPTION>
   </SELECT>
   <INPUT TYPE="submit" VALUE="Click To Submit"></INPUT>
</FORM>
</BODY>
</HTML>
```

FIG. 10C shows a web page produced using the HTML file listed above, on a display of the requesting device.

WML FILE TRANSCODED FROM THE IFML FILE WHICH INCLUDES FORM COMPONENTS

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">
```

```
<wml>
<card id="Main" title="Welcome to Joey's Test Page" newcontext="true">
<p>
   Username :<input name="textbox" type="text"></input>
   <br></br>
   Password :<input name="passwordbox" type="password"></input>
   <br></br>

   Male? :<select name="Gender">
      <option value="True">true</option>
      <option value="">false</option>
   </select><br></br>

   Please select your colour. <br></br>
   <select name="color">
      <option value="1">Blue</option>
      <option value="2">Red</option>
      <option value="3">Green</option>
   </select>

   <br></br>
   Please choose your number. <br></br>
   <select name="Number">
      <option>1</option>
      <option>2</option>
      <option>3</option>
      <option>4</option>
   </select>

   <anchor>Click To Submit
      <go href="form.jsp" method="post">
         <postfield name="textbox" value="$(textbox)"/>
         <postfield name="passwordbox" value="$(passwordbox)"/>
         <postfield name="Gender" value="$(Gender)"/>
         <postfield name="color" value="$(color)"/>
         <postfield name="Number" value="$(Number)"/>
      </go>
   </anchor>
</p>
</card>
</wml>
```

FIG. 10D shows a web page produced using the WML file listed above, on a display of the requesting device.

A further example will now be provided to illustrate how web contents are created through dynamic means. IFML handles JSP codes, which are transcoded by the IFML processor 425 into JSP for execution during runtime.

IFML FILE WHICH INCLUDES DYNAMIC CONTENTS

```
<ifml>
<head><title>JSP Capabilities Page</title></head>
<body>

<jsp>
   @ page language="java"
   import="java.util.*"
   import="java.lang.Math.*"
</jsp>

<jsp>
   int i=0;
   for (i=10;i>0;i--){
```

```
</jsp>

<jsp>
   =i*Math.rint(Math.random()*10)
</jsp>
<br/>

<jsp>
   }
</jsp>

</body>
</ifml>
```

HTML FILE TRANSCODED FROM THE IFML FILE WHICH INCLUDES DYNAMIC CONTENTS

```
<HTML>
<HEAD>
<TITLE>JSP Capabilities Page</TITLE>
</HEAD>
<BODY>
<%@
page language="java"
import="java.util.*"
import="java.lang.Math.*"
%>

<%
int i=0;
for (i=10;i>0;i--)
{
%>
<%=i*Math.rint(Math.random()*10)%><BR></BR>

<%
}
%>
</BODY>
</HTML>
```

FIG. 10E shows a web page produced using the HTML file listed above, on a display of the requesting device.

WML FILE TRANSCODED FROM THE IFML FILE WHICH INCLUDES DYNAMIC CONTENTS

```
<%@ page language="java"
  contentType="text/vnd.wap.wml" %><?xml version="1.0"?>
  <!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML
  1.3//EN" "http://www.wapforum.org/DTD/wml13.dtd">

<wml>

<card id="Main" title="JSP Capabilities Page" newcontext="true">

<p>
<%@
page language="java"
import="java.util.*"
import="java.lang.Math.*"
%>

<%
int i=0;
```

```
for (i=10;i>0;i--)
{
%>
<%=i*Math.rint(Math.random()*10)%>

<%
}
%>
</p>
</card>
</wml>
```

FIG. 10F shows a web page produced using the WML file listed above, on a display of the requesting device.

The transcoder of the present invention, as described, advantageously enables an author to design a web page once in IFML, and the IFML is transcoded to any of a variety of markup languages, dependent on the markup language required by a requesting device. As the author determines which display-components are to be author-specific, and which are to be device-specific, the author has more control as to which of the display-components will be presented in accordance with the author's specifications, and which of the display –components will be presented in accordance with respective device specifications, on a variety of devices.

This is accomplished with an authoring tool that allows an author to design a web page and produce an IFML file that is stored at a user's website, where the IFML file includes author-specific and device specific display components, in accordance with the author's design. When a web page request is received at the website, a transcoding system at the website detects the markup language requirement of the requesting device, and gets the IFML file associated with the web page requested by the requesting device. The transcoding system then transcodes the IFML file to produce the requested web page in the markup language of the requesting device, where the markup language includes the author-specific display-components and the device-specific display components. The transcoding system then provides the requested web page in the markup language of the requesting device to the requesting device, where the author-specific display-components are presented in accordance with the author's specifications, and the device-specific

display-components are presented in accordance with the receiving device's specifications.

Thus, the present invention, as described provides a method and apparatus for authoring and publishing web pages, which overcomes or at least reduces the abovementioned problems of the prior art.

It will be appreciated that although only particular embodiments of the invention have been described in detail, various modifications and improvements can be made by a person skilled in the art without departing from the scope of the present invention.

Claims

1. A transcodable formatting language for specifying the format of an electronic document, wherein the transcodable formatting language is transcodable to at least one presentation language, the at least one presentation language for presenting a transcoded version of the electronic document on at least one of a variety of devices, the transcodable formatting language comprising:

at least one author-specific display-component that is presented on the at least one of the variety of devices in accordance with author specifications; and

at least one device-specific display component that is presented on the at least one of the variety of devices in accordance with device specifications of the at least one of the variety of devices.

2. A transcodable formatting language in accordance with claim 1 wherein the electronic document comprises a web page.

3. A transcodable formatting language in accordance with claim 1 wherein the at least one presentation language comprises a markup language.

4. A transcodable formatting language in accordance with claim 1 wherein the markup language comprises at least one markup language from the group consisting of:

hypertext markup language;

wireless markup language;

extensible markup language;

extensible hypertext markup language; and

compact hypertext markup language.

5. A transcodable formatting language in accordance with claim 1 wherein the at least one author-specific display-component comprises at least one textual display-component.

6. A transcodable formatting language in accordance with claim 5 wherein the at least one textual display-component comprises at least one user-fillable form display-component.

7. A transcodable formatting language in accordance with claim 5 wherein the at least one textual display-component comprises at least one dynamic content display-component.

8. A transcodable formatting language in accordance with claim 1 wherein the at least one author-specific display-component comprises at least one image display-component.

9. A transcodable formatting language in accordance with claim 1 wherein the at least one device-specific display-component comprises at least one textual display-component.

10. A transcodable formatting language in accordance with claim 9 wherein the at least one textual display-component comprises at least one user-fillable form display-component.

11. A transcodable formatting language in accordance with claim 9 wherein the at least one textual display-component comprises at least one dynamic content display-component.

12. A transcodable formatting language in accordance with claim 1 wherein the at least one device-specific display-component comprises at least one image display-component.

13. An authoring tool for authoring an electronic document in a transcodable formatting language, wherein the transcodable formatting language is transcodable to at least one presentation language, the at least one presentation language for presenting a transcoded version of the electronic document on at least one of a variety of devices, the authoring tool comprising:

a first input for receiving at least one display-component from an author of the electronic document;

a second input for receiving layout information associated with the at least one display-component from the author of the electronic document;

a third input for receiving display-component specification associated with the at least one display-component from the author of the electronic document, the display-component specification for indicating the at least one display-component will be presented in accordance with specification provided by the author when the display-component specification is author-specific, and the display-component specification indicating the at least one display-component will be presented in accordance with device specification of the at least one of the variety of devices when the display-component specification is device-specific; and

an output for providing the electronic document in the transcodable formatting language.

14. An authoring tool in accordance with claim 13 further comprising a display-component listing area for listing the at least one display- component received from the first input, wherein the at least one display-component is presented to the author in the display-component listing area to allow the author to select the at least one display-component.

15. An authoring tool in accordance with claim 14 further comprising an author-specific selector input for the author to indicate that the at least one display-component is to be an author-specific display-component.

16. An authoring tool in accordance with claim 15 further comprising a display-component positioning area for displaying the author-specific display-component, and for allowing the author to select the position of the author-specific display-component, thereby providing layout information associated with the author-specific display-component.

17. An authoring tool in accordance with claim 14 further comprising an device-specific selector input for the author to indicate that the at least one display-component is to be a device-specific display-component.

18. An authoring tool in accordance with claim 17 further comprising a display-component positioning area for displaying the device-specific display-component, and for allowing the author to select the position of the device-specific display-component, thereby providing layout information associated with the device-specific display-component.

19. A method for authoring an electronic document in a transcodable formatting language, wherein the transcodable formatting language is transcodable to at least one presentation language, the at least one presentation language for presenting a transcoded version of the electronic document on at least one of a variety of devices, the method comprising the steps of:

a) providing at least one display-component;

b) providing display-component specification for the at least one display-component, wherein step (b) comprises the step of the author selecting between an author-specific selection and a device-specific selection;

c) providing layout information for the at least one display-component; and

d) generating the electronic document in transcodable formatting language, wherein the electronic document includes the at least one display component, the display-component specification associated with the at least one display-component, and the layout information associated with the at least one display component.

20. A method for authoring an electronic document in accordance with claim 19, wherein step (a) comprises the step of providing a plurality of display-components that includes the at least one display-component.

21. A method for authoring an electronic document in accordance with claim 20, wherein step (a) comprises the step of providing an XML file that includes the at least one display-component.

22. A method for authoring an electronic document in accordance with claim 19, wherein step (a) comprises the step of providing at least one display component from the group consisting of:

textual display-component; and

image display-component.

23. A method for authoring an electronic document in accordance with claim 19, wherein the step of the author selecting the author-specific selection comprises the author indicating the at least one display-component is an author-specific display-component.

24. A method for authoring an electronic document in accordance with claim 19, wherein the step of the author selecting the device-specific selection comprises the author indicating the at least one display-component is a device-specific display-component.

25. A method for authoring an electronic document in accordance with claim 19, wherein step (c) comprises the step of the author specifying relative position of the at least one display-component.

26. A method for authoring an electronic document in accordance with claim 25, wherein the step of the author specifying the relative position of the at least one display-component comprises the step of the author specifying a preferred relative position of the at least one display-component when the at least one display-component comprises a device-specific display-component.

27. A method for authoring an electronic document in accordance with claim 19, wherein step (d) comprises the steps of:

identifying the at least one display-component is an author-specific display-component; and

generating at least some of the transcodable formatting language for the electronic document in accordance with the display-component specifications associated with the at least one display-component.

28. A method for authoring an electronic document in accordance with claim 19, wherein step (d) comprises the steps of:

identifying the at least one display-component is an device-specific display-component; and

generating at least some of the transcodable formatting language for the electronic document in accordance with the display-component specifications associated with the at least one display-component.

29. An authoring tool service provider for providing an authoring service to at least one user, the authoring tool service provider for authoring an electronic document in a transcodable formatting language, wherein the transcodable formatting language is transcodable to at least one presentation language, the at least one presentation language for presenting a transcoded version of the electronic document on at least one of a variety of devices, the authoring tool service provider comprising:

an input coupled to receive at least one display-component, layout information for the at least one display-component, and display-component specification associated

with at least one display-component of the electronic document, the display-component specification for indicating the at least one display-component will be presented in accordance with specification provided by the at least one user when the display-component specification is author-specific, and the display-component specification indicating the at least one display-component will be presented in accordance with device specification of the at least one of the variety of devices when the display-component specification is device-specific; and

an output adapted to provide the electronic document in the transcodable formatting language to the at least one user.

METHOD AND APPARATUS FOR AUTHORING AND PUBLISHING WEB PAGES

Abstract

An author provides display-components comprising text (115) and images (120) to an authoring tool (100). The author sets each selected display-component to be either author-specific i.e. displayed on a web page in accordance with the author's specification, or device-specific i.e. displayed on a web page in accordance with the device's specification, with display-component specifications (134). The author also provides layout information (130). The authoring tool then generates an iFormat markup language (IFML) file, (135) which is stored on a server. When the server receives a web page request (430) from a user device for the IFML authored web page, a request processor (425) determines the type of requesting device, and a markup language that is compatible with the requesting device is selected. An IFML transcoder (410), then transcodes the IFML file into the selected markup language, such that the author–specific display-components and the device-specific display-components are displayed accordingly.

FIGS. 1 & 4 accompany this abstract.