Title: TAB ORDER MANAGEMENT IN A PORTAL ENVIRONMENT

Abstract: Embodiments of the present invention address deficiencies of the art in respect to managing tabbing order in a portal environment and provide a novel and non-obvious method, system and computer program product for tab order management for portlets in a portal environment. In a first embodiment, a tab order management data processing system can include a portal server configured to render a portal page, a portlet aggregator configured to provide portlet markup for different portlets in the portal page, and tab order management logic coupled to the portlet aggregator. The tab order management logic can be configured to transform a relative tab order for tabbed elements in each of the different portlets into a single tab order for the portal page. For instance, the tab order management logic can include a configuration to transform relative indices for the tabbed elements in each of the different portlets into a single index for the portal page.
ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— with international search report

— before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.
TAB ORDER MANAGEMENT IN A PORTAL ENVIRONMENT

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to tab order management of controls in a markup language environment and more particularly to the management of tab ordering in a portal environment.

Description of the Related Art

Distributing content about large computer communications networks is not without its challenges. In particular, the quantity of content available for distribution in a computer communications network often varies proportionally to the size of the computer communications network. At the extreme, the Internet hosts a vast quantity of content not easily accessible by most end-users. Portals represent a sensible solution to the problem of aggregating content through a channel paradigm in a single, network-addressable location. In consequence, portals have become the rage in content distribution.

Portlets are the visible active components included as part of portal pages. Similar to the graphical windows paradigm of windowing operating systems, each portlet in a portal occupies a portion of the portal page through which the portlet can display associated content from a portlet channel. Portlets are known to include both simple applications such as an electronic mail client, and also more complex applications such as forecasting output from a customer relationship management system. The prototypical portlet can be implemented as server-side scripts executed through a portal server.

From the end-user perspective, a portlet is a content channel or application to which the end-user can subscribe. By comparison, from the perspective of the content provider, a portlet is a means through which content can be distributed in a personalized manner to a subscribing end-user. Finally, from the point of view of the portal, a portlet merely is a component which can be rendered within the portal page. In any case, by providing one or more individually selectable and configurable portlets in a portal, portal providers can distribute content and applications
through a unified interface in a personalized manner according to the preferences of the end-user.

Portal servers are computer programs which facilitate the distribution of portal based Web sites on the public Internet or a private intranet. Importantly, it will be recognized by one of ordinary skill in the art that the signature characteristic of all conventional portal servers can include the aggregation of content from several portlet applications within a single distributable page in a uniform manner. To that end, each portlet application within the portal page can be represented by a portlet user interface distributed by the portal server to requesting client computing devices.

The aggregation of content from several portlet applications within a single distributable page in a uniform manner can present unique challenges in crafting a navigable user interface. Specifically, in an ordinary user interface, setting the tabbing order for navigating between elements in the user interface can be a matter of establishing the tabbing order through an indexing scheme. In a portal environment, however, the indexing scheme of any one portlet can conflict with the indexing scheme of the other portlets. Moreover, unlike a conventional user interface which usually is within the control of a single person or team of persons, in a portal environment, literally each portlet can be within the control of different persons. Accordingly, it will not be reasonable to expect coordination among the different persons in establishing a proper tabbing order for elements in each portlet.

To explain the above in more detail, when not using the mouse, people typically navigate through form controls with the tab key. If tabbing order is not specified, more than likely the browser will simply present the tabbing sequence order as the order in which the form controls appear in the source code of the HTML. This could cause the tabbing to appear haphazard and perhaps a little difficult to follow and understand. Specifying a tab order allows them to navigate through these items in a logical order. In HTML, "tabIndex" attribute specifies the tabbing navigation order among form controls. The browser will tab to the elements in ascending order. This helps solve the problem of "difficult" tabbing within a page.

However, the above solution is only applicable when the HTML writer owns the entire page. It falls a little short in environments where different HTML outputs are brought together and rendered on the same page. In some cases, the portlets are written independently of each other and
have no concept of what portlets they are going to share a portal page with. There is the potential for "tabindex" attributes specified in one portlet to conflict with those specified in another. Also there is the possibility of portlets sharing a page where some have specified tabbing order, while others haven't. Therefore the problem arises again where tabbing can appear haphazard and perhaps a little difficult to follow and understand.

Take the following example of a portal page:

<table>
<thead>
<tr>
<th>Portlet A</th>
<th>Portlet B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portlet C</td>
<td>Portlet D</td>
</tr>
</tbody>
</table>

Logically a user may expect to tab in the order of (reading order):

Portlet A ---> Portlet B ---> Portlet C ---> Portlet D

Portlet A has a snippet of code like the following:

Text A1: `<INPUT type="text" tabIndex="1"/>`
Text A2: `<INPUT type="text" tabIndex="2"/>`

Portlet B has no code that has tabbing order
Portlet C has no code that has tabbing order

Portlet D has a snippet of code like the following:

Text D1: `<INPUT type="text" tabIndex="1"/>`
Text D2: `<INPUT type="text" tabIndex="2"/>`

Obviously, there is a conflict with the tab index used. These conflicts could most likely be a result of the fact that the portlets were written independently of each other.

The browser will tab to the elements in ascending order. Therefore the user would most likely see tabbing like the following.

A1 in Portlet A ---> D1 in Portlet D ---> A2 in Portlet A ---> D2 in Portlet D ---> Then the default tab order for Portlet B and Portlet C

This is not a desired or logical tabbing sequence for the user.
To get around this issue, one might suggest not using the "tabIndex" attribute at all, and let the browser simply tab most likely using the order in which the focusable components appear in the source. But once again, there is no guarantee that will be a logical tabbing order for the user.

**BRIEF SUMMARY OF THE INVENTION**

Embodiments of the present invention address deficiencies of the art in respect to managing tabbing order in a portal environment and provide a novel and non-obvious method, system and computer program product for tab order management for portlets in a portal environment.

According to a first aspect, there is provided a tab order management data processing system comprising a portal server configured to render a portal page, a portlet aggregator configured to provide portlet markup for different portlets in the portal page, and tab order management logic coupled to the portlet aggregator. The tab order management logic is configured to transform a relative tab order for tabbed elements in each of the different portlets into a single tab order for the portal page.

According to one embodiment, the tab order management logic can include a configuration to transform relative indices for the tabbed elements in each of the different portlets into a single index for the portal page.

According to a second aspect, there is provided a tab order management method comprising transforming a relative tab order for tabbed elements in each of different portlets into a single tab order for a portal page in a portal environment.

In one embodiment, transforming a relative tab order for tabbed elements in each of different portlets into a single tab order for a portal page in a portal environment, can include transforming each relative index value for a relative index for tabbed elements in a portlet into an absolute index value for a uniform index for the portal page.

In one embodiment, transforming a relative tab order for tabbed elements in each of different portlets into a single tab order for a portal page in a portal environment comprises, selecting a portlet in the
portal page, adding a current index value for a uniform index for the portal page to each relative index value for each tabbed element in the selected portlet to produce a uniform index value for the tabbed element, changing the current index value to a value accounting for a last processed tabbed element in the selected portlet, and repeating the retrieving and adding for each additional portlet in the portal page.

Additional aspects of the invention will be set forth in part in the description which follows, and in part will be obvious from the description, or may be learned by practice of the invention. The aspects of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims. It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention, as claimed.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

Preferred embodiments of the present invention will now be described, by way of example only, and with reference to the following drawings:

Figure 1 is a schematic illustration of a portal environment data processing system configured for tab order management in accordance with a preferred embodiment of the present invention; and,

Figure 2 is a flow chart illustrating a process for tab order management in a portal environment in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Embodiments of the present invention provide a method, system and computer program product for tab order management in a portal environment. In accordance with an embodiment of the present invention, the tab order for elements within different portlets in the portal environment can be unified into a single sequence. Specifically, the sequence of display of the portlets in the portal environment can be used to transform the relative ordering of elements within each portlet into a single ordering of elements in the portal display. In this way, conflicting indexing schemes for the different portlets can be unified in the portal display.
In more particular illustration, Figure 1 is a schematic illustration of a portal environment data processing system configured for tab order management. The portal environment data processing system can include a portal page 135 communicatively coupled to a selection ofportlet applications 115A, 115B, 115n through a portal server 130. Each portlet application 115A, 115B, 115n can produce a view based upon portlet data 120A, 120B, 120n in the form of portlet markup 125 through corresponding rendering logic HOA, HOB, HOn. Notably, the rendering logic HOA, HOB, HOn can be active markup such as a JSP, in which logical scriptlets can be embedded to produce specific markup language tags.

A portlet aggregator 105 can be coupled to each portlet application 115A, 115B, 115n to receive the portlet markup 125 and to aggregate the portlet markup 125 into view in the portal page 135. By aggregation, it is meant that the individual markup language blocks produced by each portlet 115A, 115B, 115n can be combined into a single cohesive markup language document configured for distribution to and use within a conventional content browser. In this regard, the portal page 135 can be disposed in the portal server 130 from which the portal 135 can be accessed by client content browsing devices 140 over a computer communications network 160 such as a local computer communications network, for instance a private intranet, or a global computer communications network, for instance the public Internet.

Notably, tab order management logic 200 can be coupled to the aggregator 105. The tab order management logic 200 can be configured to transform the relative tab order indices of each tabbed element in each portlet 115B into a single, unified tab order index. In more particular illustration, Figure 2 is a flow chart illustrating a process for tab order management in a portal environment. Beginning in block 210, an index for the portal display can be set to an initial value. In block 220, a first portlet can be selected for processing. In block 230, the number of tabbed elements in the selected portlet can be counted. Specifically, each element which permits tabbed navigation can be counted to determine a number of elements which can be navigated through the activation of a tab operation.

In block 240, the relative tab order for the selected portlet can be transformed into a uniform tab order for the portal page in which the
portlet is to be displayed. In this regard, the transformation of the relative tab order for the selected portlet can account for the order of the portlets in the portal display and the number of tabbed elements already sequenced for other portlets. Specifically, for n processed portlets having m tabbed elements among the n processed portlets, the index of the first tabbed element in the n+1 portlet having x tabbed elements will be m+1 and the index of the last tabbed element in the n+1 portlet will be n+x.

Hence, in block 240, the relative index for each tabbed element in the selected portlet can be transformed to a uniform index having a value resulting from the addition of the relative index value and the current index value for the portal display. Subsequently, in decision block 250, if additional portlets remain to be processed, in block 260, the current index value for the portal display can be set to the transformed index value for the last tabbed element in the selected portlet. Thereafter, the next portlet can be retrieved in block 270 and the process can continue through block 230. When no further portlets remain to be processed, the transformation process can end and the portal display can be rendered in block 280.

To summarise, the solution disclosed herein proposes the idea of correlating portlet position on a page with that of the tabbing order within a portlet. This helps to facilitate sequential and logical tabbing order for the user. This is achieved through the following steps:

1) Determine Portlet Position

2) Determine the largest index in a portlet and add to the tab indices of the next sequentially positioned portlet.

This ensures correct order between portlets. It also keeps the integrity of the already established tab indices, so portlets will have the same tabbing order as coded.

3) A portal rendering engine (which ultimately owns the final client response, or HTML Page) preferably incorporates and re-writes all applicable tab indices in each Portlet to create a unified tab index representation in order to provide a cohesive user interaction behaviour.

To explain in more detail, take the following portal page:
More than likely the tabbing order (without the solution disclosed herein) would be as follows:

Portlet A

Text A1: `<INPUT type="text" tabIndex="1"/>

Text A2: `<INPUT type="text" tabIndex="2"/>

Portlet B

Text B1: `<INPUT type="text" tabIndex="1"/>

Text B2: `<INPUT type="text" tabIndex="2"/>

Portlet C

Text C1: `<INPUT type="text" tabIndex="1"/>

Text C2: `<INPUT type="text" tabIndex="2"/>

Portlet D

Text D1: `<INPUT type="text" tabIndex="1"/>

Text D2: `<INPUT type="text" tabIndex="2"/>

More than likely the tabbing order (without the solution disclosed herein) would be as follows:

A1 in Portlet A ---> B1 in Portlet B ---> C1 in Portlet C ---> D1 in Portlet D ---> A2 in Portlet A ---> B2 in Portlet B ---> C2 in Portlet C ---> D2 in Portlet D

With the proposal disclosed above the results would be as follows:

Portlet A

Text A1: `<INPUT type="text" tabIndex="1"/>

Text A2: `<INPUT type="text" tabIndex="2"/>

Portlet A is the first portlet and has implemented tab indices. These indices will not be changed. The greatest tab index is 2.

Portlet B

Text B1: `<INPUT type="text" tabIndex="3"/>

Text B2: `<INPUT type="text" tabIndex="4"/>

The largest tab index in Portlet A is 2. Therefore 2 is added to each of the tab indices in Portlet B to get the above tab indices

Portlet C

Text C1: `<INPUT type="text" tabIndex="5"/>
Text C2: <INPUT type="text" tabIndex="6"/>

The largest tab index in Portlet B is 4. Therefore 4 is added to each of the tab indices in Portlet C to get the above tab indices.

Portlet D

Text D1: <INPUT type="text" tabIndex="7"/>
Text D2: <INPUT type="text" tabIndex="8"/>

The largest tab index in Portlet C is 6. Therefore we add 6 to each of the tab indices in Portlet D to get the above tab indices.

The tabbing order would therefore be:

A1 in Portlet A ----> A2 in Portlet A ----> B1 in Portlet B ----> B2 in Portlet B ----> C1 in Portlet C ----> C2 in Portlet C ----> D1 in Portlet D ----> D2 in Portlet D

Embodiments of the invention can take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment containing both hardware and software elements. In a preferred embodiment, the invention is implemented in software, which includes but is not limited to firmware, resident software, microcode, and the like. Furthermore, the invention can take the form of a computer program product accessible from a computer-readable medium providing program code for use by or in connection with a computer or any instruction execution system.

For the purposes of this description, a computer-readable medium can be any apparatus that can contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device. The medium can be an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system (or apparatus or device) or a propagation medium. Examples of a computer-readable medium include a semiconductor or solid state memory, magnetic tape, a removable computer diskette, a random access memory (RAM), a read-only memory (ROM), a rigid magnetic disk and an optical
Current examples of optical disks include compact disk - read only memory (CD-ROM), compact disk - read/write (CD-R/W) and DVD.

A data processing system suitable for storing and/or executing program code will include at least one processor coupled directly or indirectly to memory elements through a system bus. The memory elements can include local memory employed during actual execution of the program code, bulk storage, and cache memories which provide temporary storage of at least some program code in order to reduce the number of times code must be retrieved from bulk storage during execution. Input/output or I/O devices (including but not limited to keyboards, displays, pointing devices, etc.) can be coupled to the system either directly or through intervening I/O controllers. Network adapters may also be coupled to the system to enable the data processing system to become coupled to other data processing systems or remote printers or storage devices through intervening private or public networks. Modems, cable modem and Ethernet cards are just a few of the currently available types of network adapters.
CLAIMS

1. A tab order management data processing system comprising:

   a portal server configured to render a portal page;

   a portlet aggregator configured to provide portlet markup for different portlets in said portal page; and,

   tab order management logic coupled to said portlet aggregator and configured to transform a relative tab order for tabbed elements in each of said different portlets into a single tab order for said portal page.

2. The data processing system of claim 1, wherein said tab order management logic comprises a configuration to transform relative indices for said tabbed elements in each of said different portlets into a single index for said portal page.

3. The data processing system of claim 1, wherein the tab order management logic is operable to transform each relative index value for a relative index for tabbed elements in a portlet into an absolute index value for a uniform index for said portal page.

4. The data processing system of claim 1, the tab order management logic further comprises:

   means for selecting a portlet in said portal page;

   means for adding a current index value for a uniform index for said portal page to each relative index value for each tabbed element in said selected portlet to produce a uniform index value for said tabbed element;

   means for changing said current index value to a value accounting for a last processed tabbed element in said selected portlet; and,

   means for repeating said retrieving and adding for each additional portlet in said portal page.

5. A tab order management method comprising transforming a relative tab order for tabbed elements in each of different portlets into a single tab order for a portal page in a portal environment.
6. The method of claim 5, wherein said transforming a relative tab order for tabbed elements in each of different portlets into a single tab order for a portal page in a portal environment, comprises transforming each relative index value for a relative index for tabbed elements in a portlet into an absolute index value for a uniform index for said portal page.

7. The method of claim 5, wherein said transforming a relative tab order for tabbed elements in each of different portlets into a single tab order for a portal page in a portal environment, comprises:

   selecting a portlet in said portal page;

   adding a current index value for a uniform index for said portal page to each relative index value for each tabbed element in said selected portlet to produce a uniform index value for said tabbed element;

   changing said current index value to a value accounting for a last processed tabbed element in said selected portlet; and,

   repeating said retrieving and adding for each additional portlet in said portal page.

8. A computer program product comprising a computer usable medium having computer usable program code for tab order management in a portal environment, said computer program product including computer usable program code for transforming a relative tab order for tabbed elements in each of different portlets into a single tab order for a portal page in a portal environment.

9. The computer program product of claim 8, wherein said computer usable program code for transforming a relative tab order for tabbed elements in each of different portlets into a single tab order for a portal page in a portal environment, comprises computer usable program code for transforming each relative index value for a relative index for tabbed elements in a portlet into an absolute index value for a uniform index for said portal page.

10. The computer program product of claim 8, wherein said computer usable program code for transforming a relative tab order for tabbed elements in each of different portlets into a single tab order for a portal page in a portal environment, comprises:
computer usable program code for selecting a portlet in said portal page;

computer usable program code for adding a current index value for a uniform index for said portal page to each relative index value for each tabbed element in said selected portlet to produce a uniform index value for said tabbed element;

computer usable program code for changing said current index value to a value accounting for a last processed tabbed element in said selected portlet; and,

computer usable program code for repeating said retrieving and adding for each additional portlet in said portal page.

11. A computer program comprising program code means adapted to perform the method of any of claims 5 to 7 when said program is run on a computer.
FIG. 1
FIG. 2

Set Index := 1

Get First Portlet

Get Next Portlet

Get Element Count

Set Tab Order(Index)

Set Index := Count

More?

YES

NO

Render Page
INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2006/064147

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
S06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and where practical search terms used)
EPO-Internal, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category’ Citation of document, with indication, where appropriate, of the relevant passages Relevant to claim No

abstract figures 3C, 5A
column 1, lines 30-40
column 2, lines 1-8
column 5, lines 59-65
column 8, lines 37-58
column 9, line 56 - column 10, line 18

Further documents are listed in the continuation of Box C See patent family annex

Date of the actual completion of the international search
16 October 2006

Date of mailing of the international search report
24/11/2006

Name and mailing address of the ISA/
European Patent Office, P B 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel (+31-70) 340-2040, Tx 31 651 epo nl
Fax (+31-70) 340-3016

Authorized officer
Bykowski, Artur
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>RAGGETT D ET AL: &quot;HTML 4.0 Specification, CHAPTER 17, FORMS&quot;</td>
<td>1-11</td>
</tr>
<tr>
<td></td>
<td>INTERNET CITATION, [Online]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>24 April 1998 (1998-04-24), pages 219-250,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XP002306842</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Retrieved from the Internet:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[retrieved on 2004-11-22]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>page 241, line 20 - page 242, line 37</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>PONZO J ET AL: &quot;On demand Web-client technologies&quot;</td>
<td>1-11</td>
</tr>
<tr>
<td></td>
<td>IBM SYSTEMS JOURNAL IBM USA,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>vol. 43, no. 2, 2004, pages 297-315,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XP002403076</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ISSN: 0018-8670</td>
<td></td>
</tr>
<tr>
<td></td>
<td>page 299, left-hand column, last line - right-hand column, line 9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>figure 9</td>
<td></td>
</tr>
<tr>
<td>Patent document cited in search report</td>
<td>Publication date</td>
<td>Patent family member(s)</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-----------------</td>
<td>--------------------------</td>
</tr>
</tbody>
</table>