US 20110314424A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2011/0314424 A1**

**GUSMORINO et al.** (43) **Pub. Date: Dec. 22, 2011**

(54) **SCALING TYPE OVERLAY ICONS**

(75) Inventors: **PAUL A. GUSMORINO**, SEATTLE, WA (US); **TYLER K. BEAM**, REDMOND, WA (US); **RAYMOND J. CHEN**, REDMOND, WA (US); **CHARLES CUMMINS**, SEATTLE, WA (US); **ROBERT F. DAY**, BELLEVUE, WA (US); **COLIN R. ANTHONY**, BOTHELL, WA (US); **JENNY T. LAM**, SEATTLE, WA (US); **ISSAC A. SHELDON**, SEATTLE, WA (US); **DAVID G. DE VORCHIK**, SEATTLE, WA (US)

(73) Assignee: **MICROSOFT CORPORATION**, REDMOND, WA (US)

(21) Appl. No.: **13/166,590**

(22) Filed: **Jun. 22, 2011**

**Related U.S. Application Data**

(63) Continuation of application No. 11/186,827, filed on Jul. 22, 2005, now Pat. No. 7,992,103, Continuation-in-part of application No. 10/831,145, filed on Apr. 26, 2004, now abandoned.

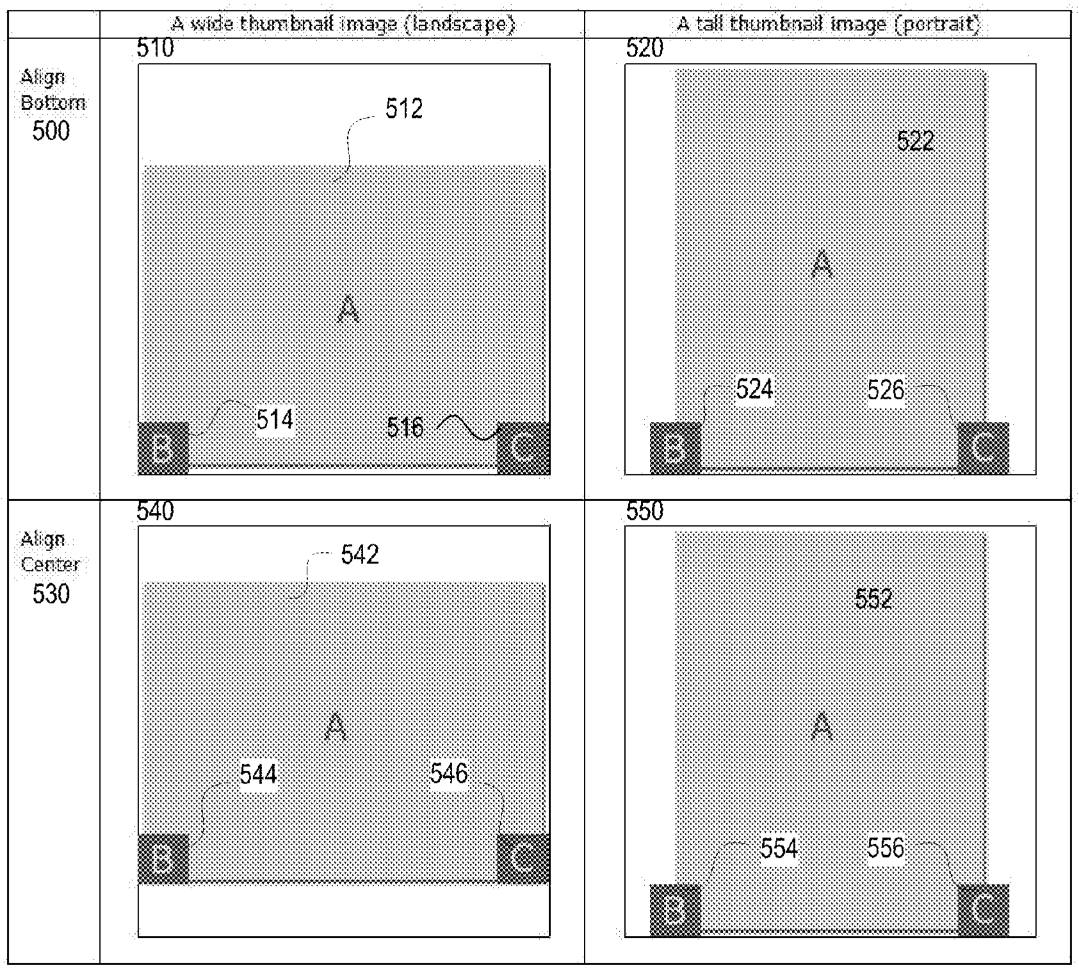
**Publication Classification**

(51) **Int. Cl.**
*G06F 3/048* (2006.01)

(52) **U.S. Cl.** ........................................................ **715/846**

(57) **ABSTRACT**

A method and system are provided for displaying icons in accordance with user preferences. The method includes providing a user with a size selection mechanism for selecting an icon size from a range of sizes. The method additionally includes searching a set of stored resources related to the underlying file and generating the icon in the selected size based upon the stored resources. The method may additionally include selecting a type overlay and scaling the type overlay independently from the icon.



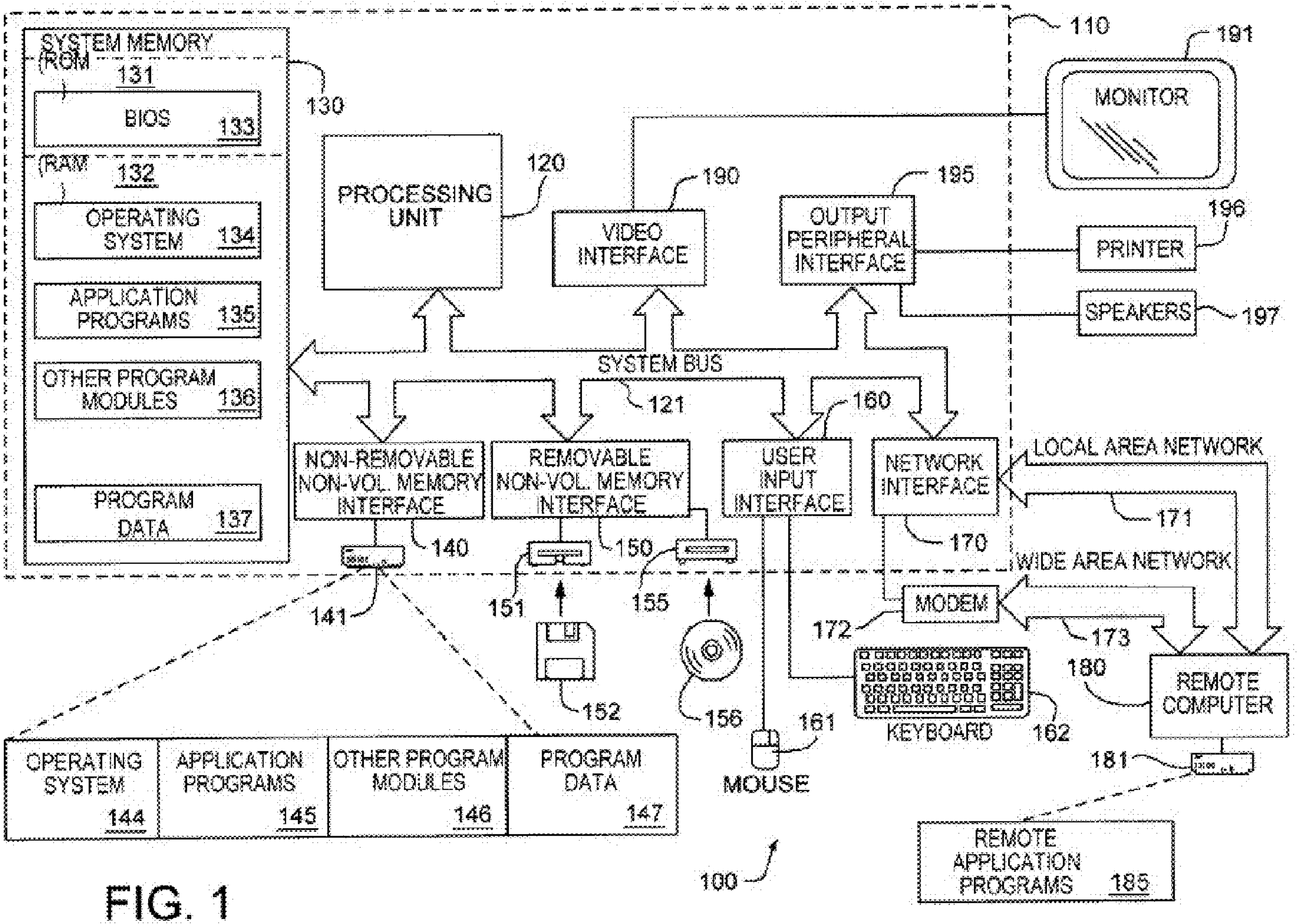

110
SYSTEM MEMORY
(ROM
131
BIOS
133
(RAM
132
OPERATING SYSTEM
134
APPLICATION PROGRAMS
135
OTHER PROGRAM MODULES
136
PROGRAM DATA
137
130
PROCESSING UNIT
120
VIDEO INTERFACE
190
OUTPUT PERIPHERAL INTERFACE
195
MONITOR
191
PRINTER
196
SPEAKERS
197
SYSTEM BUS
121
NON-REMOVABLE NON-VOL. MEMORY INTERFACE
140
REMOVABLE NON-VOL. MEMORY INTERFACE
150
USER INPUT INTERFACE
160
NETWORK INTERFACE
170
LOCAL AREA NETWORK
171
WIDE AREA NETWORK
MODEM
172
173
141
151
155
152
156
161
MOUSE
KEYBOARD
162
180
REMOTE COMPUTER
181
REMOTE APPLICATION PROGRAMS
185
OPERATING SYSTEM
144
APPLICATION PROGRAMS
145
OTHER PROGRAM MODULES
146
PROGRAM DATA
147
100

FIG. 1

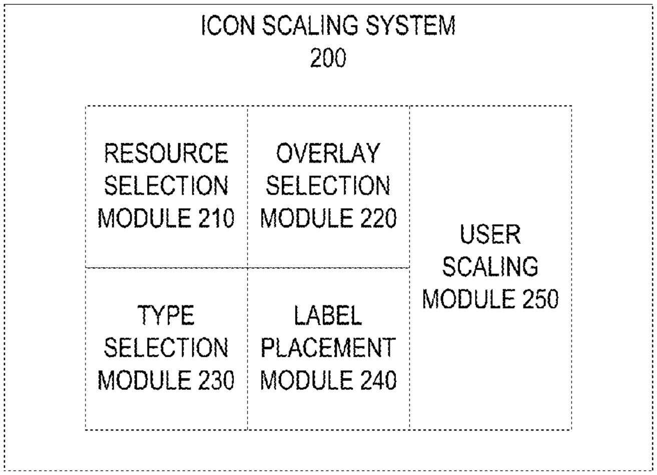

ICON SCALING SYSTEM
200
RESOURCE SELECTION MODULE 210
OVERLAY SELECTION MODULE 220
USER SCALING MODULE 250
TYPE SELECTION MODULE 230
LABEL PLACEMENT MODULE 240

*FIG. 2*

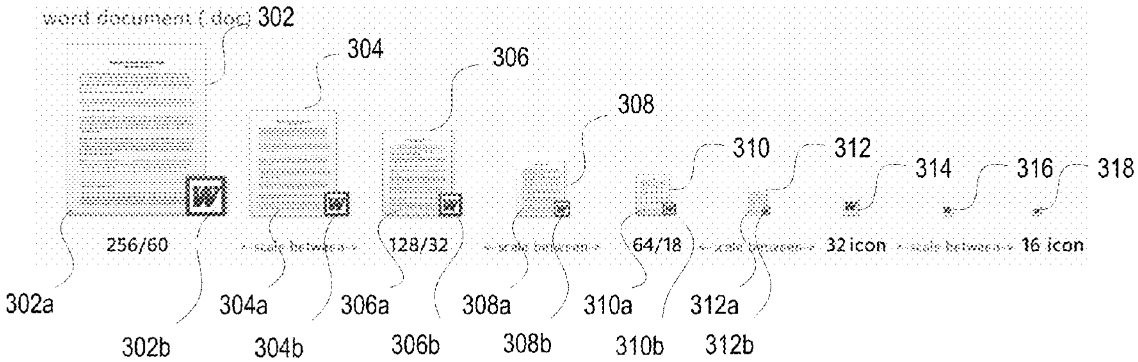

word document (.doc)
302
304
306
308
310
312
314
316
318
256/60
<- scale between ->
128/32
<- scale between ->
64/18
<- scale between ->
32 icon
<- scale between ->
16 icon
302a
302b
304a
304b
306a
306b
308a
308b
310a
310b
312a
312b

*FIG. 3A*

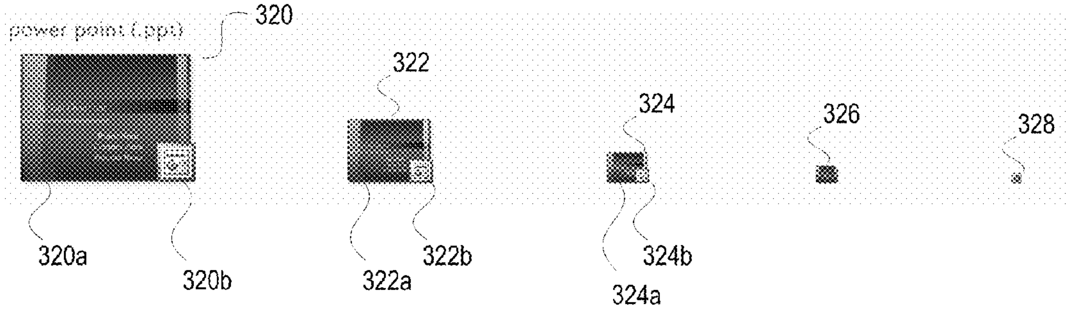


*FIG. 3B*

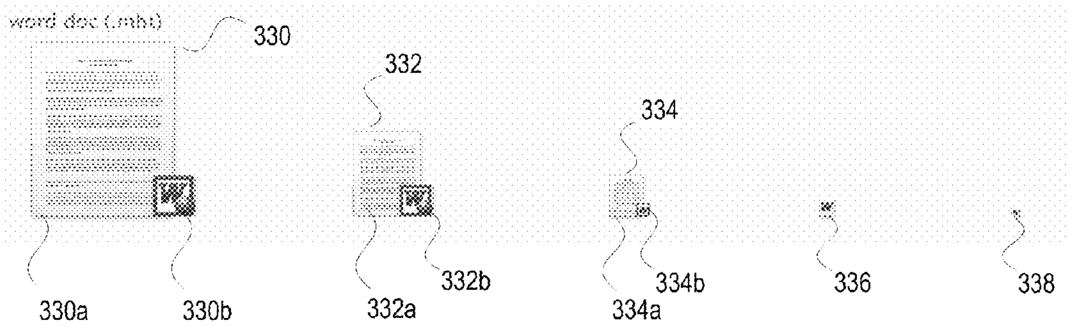


*FIG. 3C*

photos
340
342
344
346
348

*FIG. 3D*

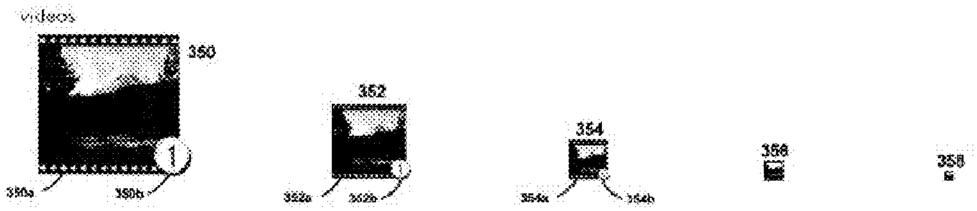


FIG. 3E

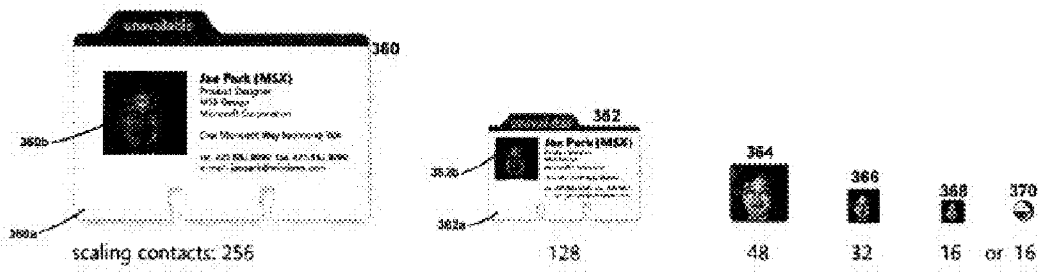


FIG. 3F

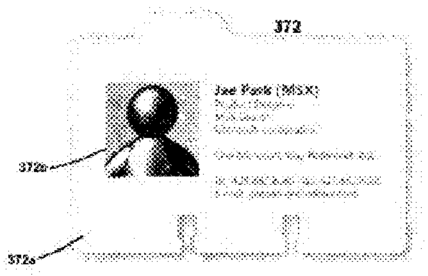


FIG. 3G

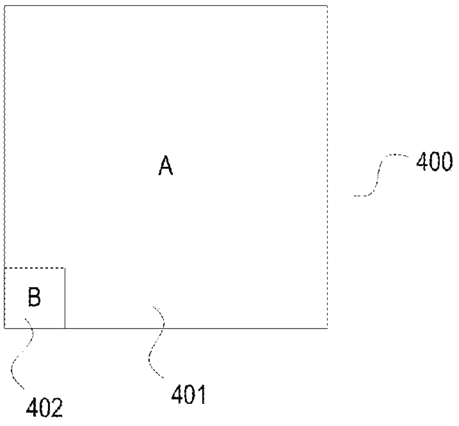

A
400
B
402
401

*FIG. 4*

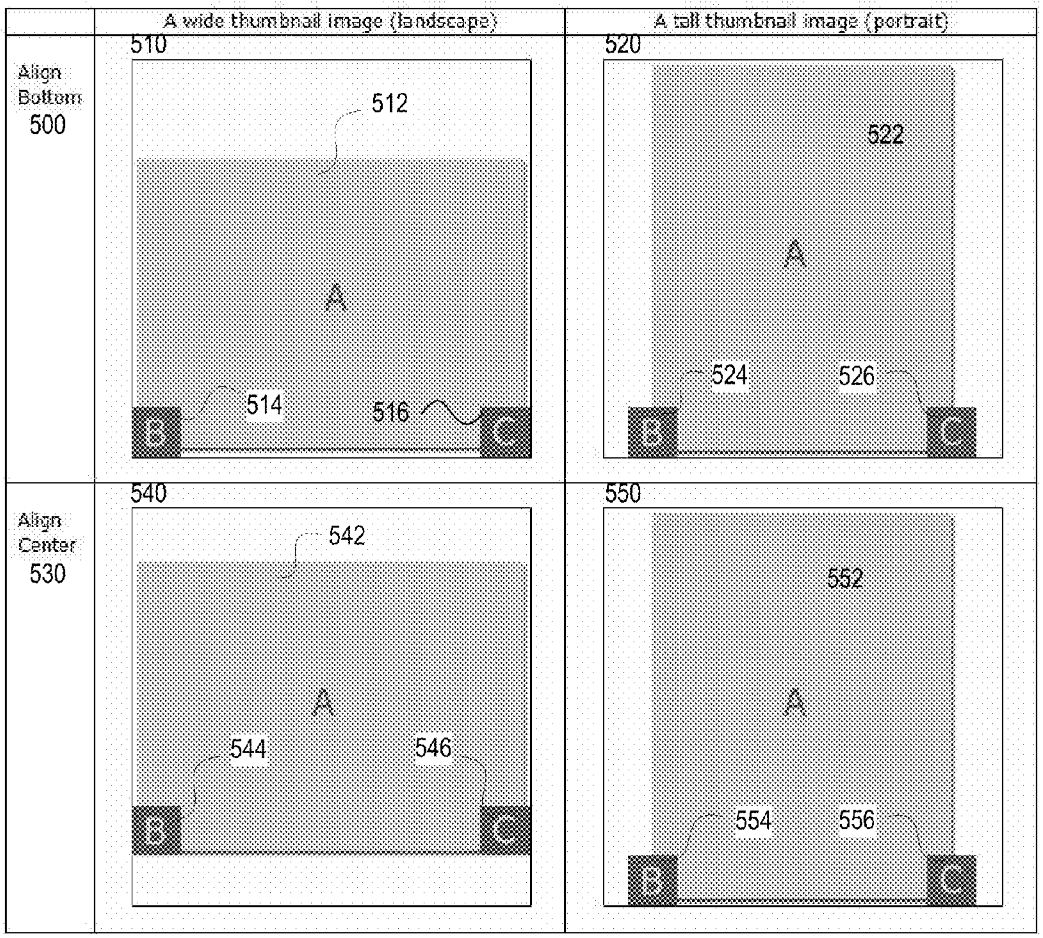

A wide thumbnail image (landscape)
A tall thumbnail image (portrait)
Align Bottom
500
510
512
A
514
516
B
C
520
522
A
524
526
B
C
Align Center
530
540
542
A
544
546
B
C
550
552
A
554
556
B
C

*FIG. 5*

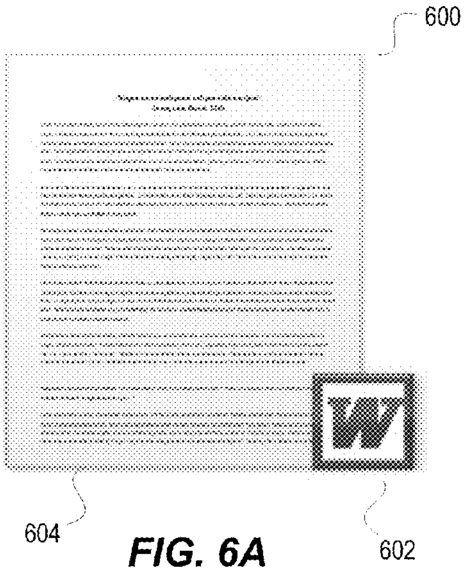


***FIG. 6A***

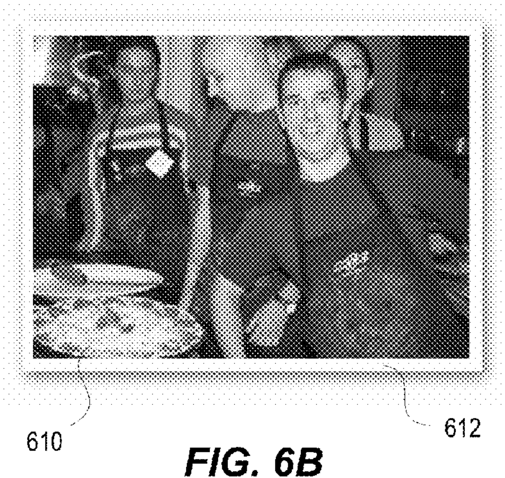


***FIG. 6B***

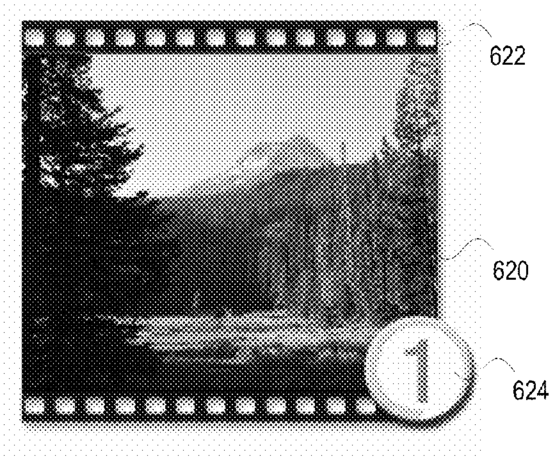


***FIG. 6C***

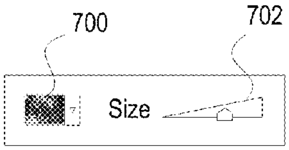


***FIG. 7***

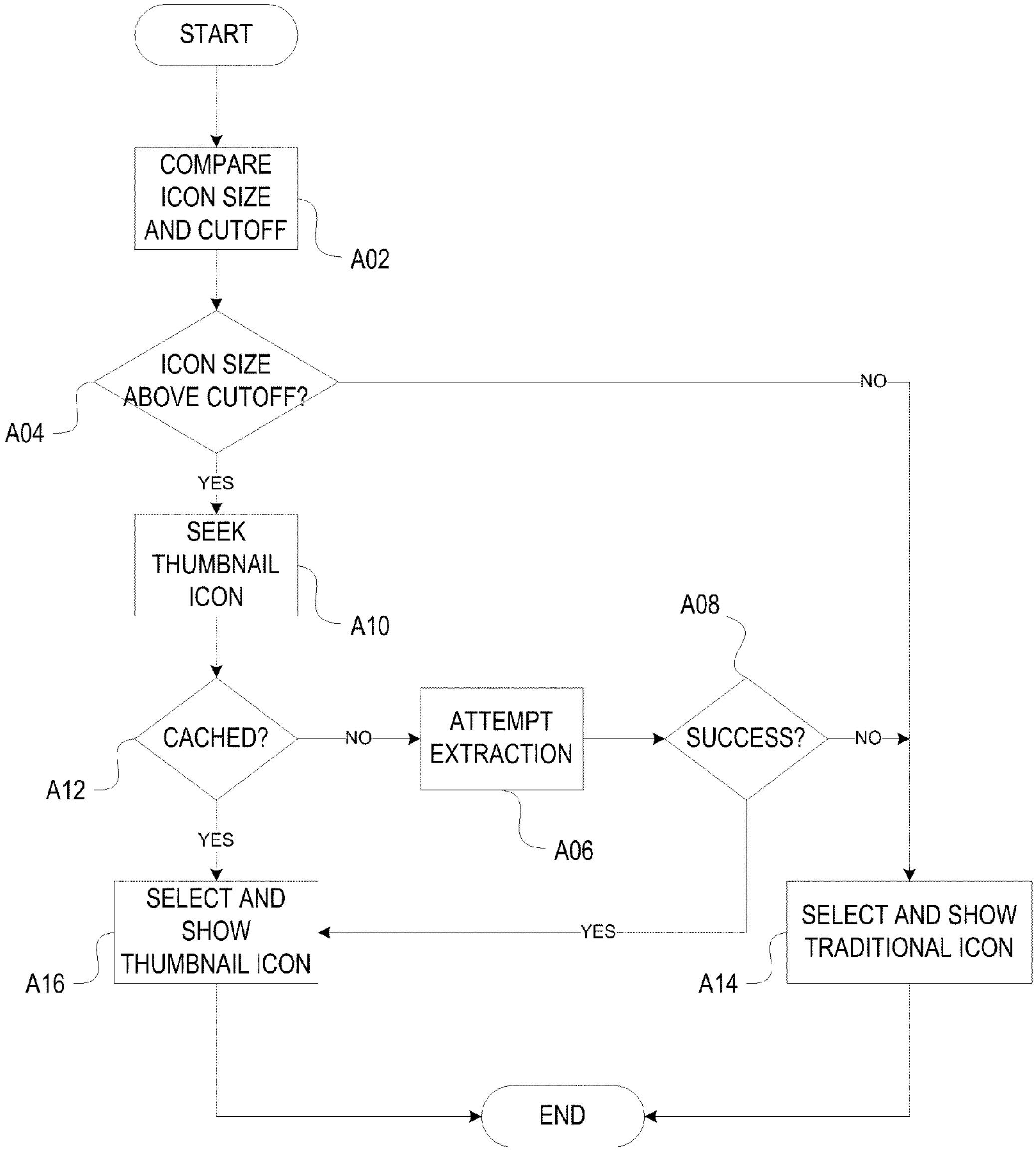

START
COMPARE ICON SIZE AND CUTOFF
A02
ICON SIZE ABOVE CUTOFF?
A04
NO
YES
SEEK THUMBNAIL ICON
A10
A08
CACHED?
A12
NO
ATTEMPT EXTRACTION
A06
SUCCESS?
NO
YES
YES
SELECT AND SHOW THUMBNAIL ICON
A16
SELECT AND SHOW TRADITIONAL ICON
A14
END

FIG. 8

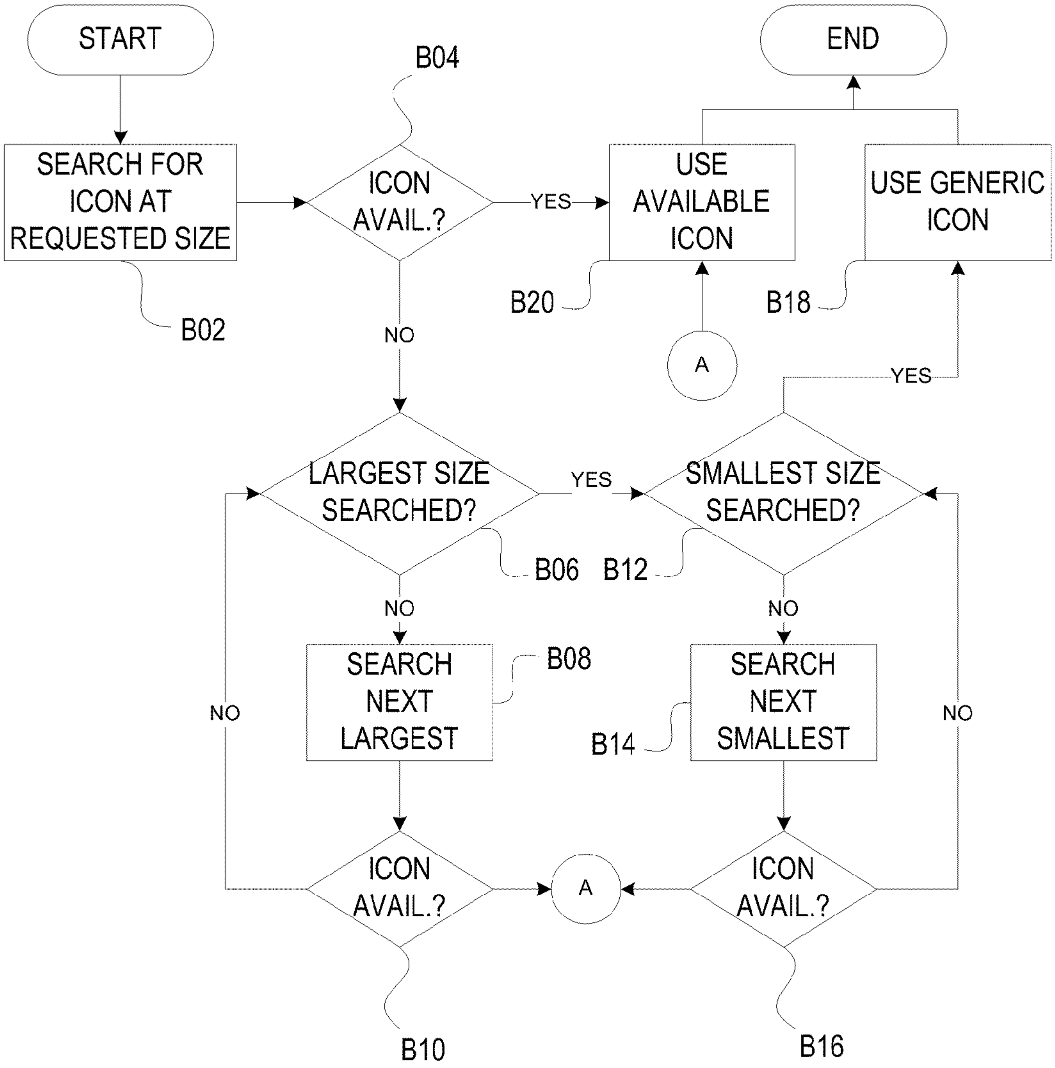

START
END
SEARCH FOR ICON AT REQUESTED SIZE
B02
B04
ICON AVAIL.?
YES
NO
USE AVAILABLE ICON
B20
USE GENERIC ICON
B18
A
YES
LARGEST SIZE SEARCHED?
YES
SMALLEST SIZE SEARCHED?
B06
B12
NO
NO
SEARCH NEXT LARGEST
B08
SEARCH NEXT SMALLEST
B14
NO
NO
ICON AVAIL.?
A
ICON AVAIL.?
B10
B16

***FIG. 9***

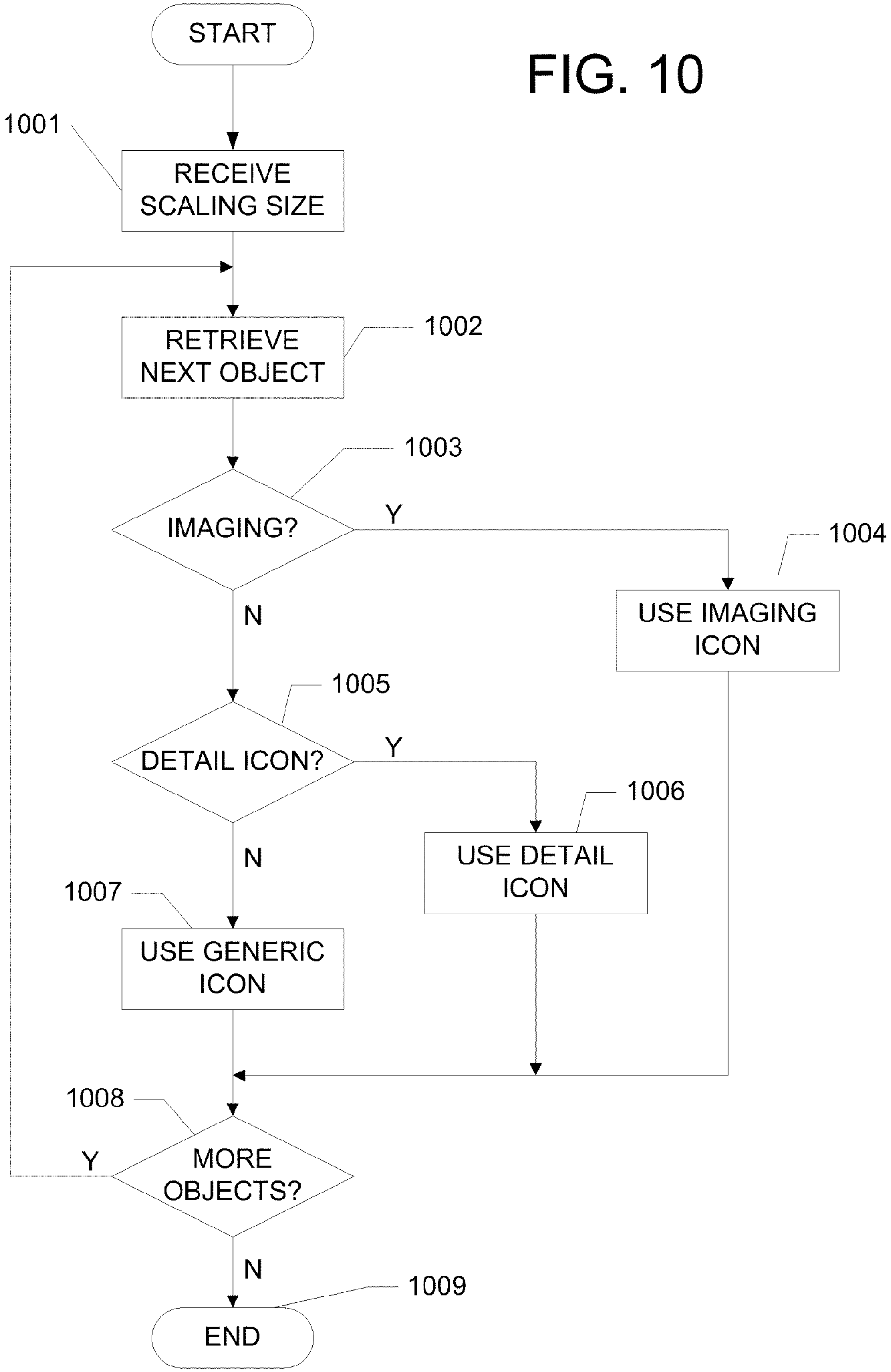

START
1001
RECEIVE SCALING SIZE
RETRIEVE NEXT OBJECT
1002
1003
IMAGING?
Y
1004
N
USE IMAGING ICON
1005
DETAIL ICON?
Y
1006
N
USE DETAIL ICON
1007
USE GENERIC ICON
1008
Y
MORE OBJECTS?
N
1009
END

FIG. 10

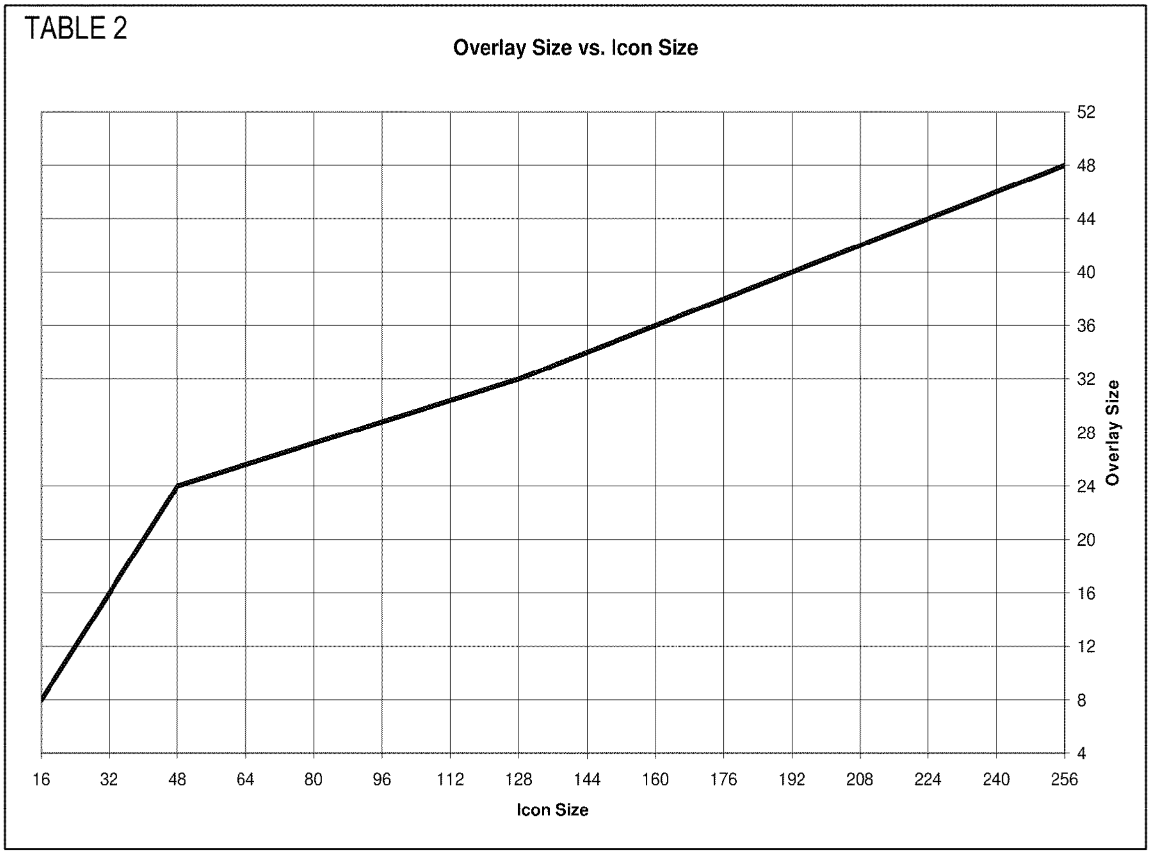


TABLE 2
Overlay Size vs. Icon Size
52
48
44
40
36
32
28
24
20
16
12
8
4
Overlay Size
16
32
48
64
80
96
112
128
144
160
176
192
208
224
240
256
Icon Size

*FIG. 11*

# SCALING TYPE OVERLAY ICONS

## CROSS-REFERENCE TO RELATED APPLICATIONS

**[0001]** This application is a continuation of copending U.S. patent application Ser. No. 11/186,827, filed Jul. 22, 2005, entitled "Scaling Icons for Representing Files," which is a continuation-in-part of U.S. patent application Ser. No. 10/831,145, filed Apr. 26, 2004, and entitled "System and Method for Scaling Icons," the entirety of each are hereby incorporated by reference.

## BACKGROUND

**[0002]** Icons have been used for many years in computer systems to display information to users. Icons are images that are used to represent items such as files, folders, shortcuts, applications, documents, tasks, and commands throughout a computer operating system such as MICROSOFT WINDOWS™, a product of the Microsoft Corporation of Redmond, Washington. Icons are currently used in toolbars, dialog boxes, menus, and numerous other locations in file management systems such as Windows Explorer, also a product of the Microsoft Corporation. Icons are usually stored as static bitmaps, but dynamic icon handlers may also be written to generate icons dynamically.

**[0003]** Currently, two types of icons are used to represent items. These two types include traditional icons and thumbnail icons. Whereas traditional icons are typically shared across files and are determined per type, thumbnails are determined per item and are not shared across files. A traditional icon is composed of two components including an icon image and a status overlay. The icon image forms the basis of the traditional icon. The status overlay is used to convey status information about the item. It is used, for example, to display the shortcut overlay.

**[0004]** A thumbnail icon is a particular type of icon displaying an image generated based on the contents of the underlying item. For example, an icon for an image file that actually looks like the image itself is a thumbnail of the image file. Thumbnail icons may also include an overlay. Thumbnail icons may be dynamically generated by code run on a user computer, although the generated image may be cached.

**[0005]** Although existing operating systems offer thumbnails, these thumbnails are not well integrated into existing file management systems such as Windows Explorer. As a result, the thumbnail feature is not as useful as it could be. A user has to go to a special "thumbnails view" to see thumbnail icons today, even for objects having thumbnail icons that would be visually effective at a smaller size, such as photographs. When in the thumbnails view of existing systems, icons for objects without thumbnails appear small and inadequate. The user typically cannot see extra information about each file in thumbnails view as the user can in other views such as tile view or details view. Accordingly, the user is often forced to choose between viewing thumbnails and viewing other important information.

**[0006]** Because the thumbnails feature is remote and not extensively utilized, little incentive exists for Independent Software Vendors (ISVs) to write thumbnail generators, When ISVs do write the thumbnail generators, they are unlikely to invest in making the thumbnail generators powerful. As a result, many existing application produce thumbnails of poor quality. Generally, since thumbnails are pictures, it is difficult to determine what type of item underlies a thumbnail icon. Furthermore, while traditional icons appear instantly, thumbnail icons are typically noticeably slow to appear because they were not generated in advance.

**[0007]** With traditional icons, overlays are often used as a primary or only indication of information about an item underlying an icon. However, overlays do not reliably appear. If multiple states need to be shown, overlays will only show the highest priority state and will ignore others since no more than one overlay may be shown at a time. Overlays are difficult to recognize at small icon sizes, such as the sizes generally used in list or details mode. Overlays have additional drawbacks including the ability to be readily imitated and used in an unauthorized manner. A further limitation is the inaccessibility of overlays to the visually-impaired. A user navigating with the keyboard and using a narrator hears a name and type of each file as the file is selected, but is not given any information about a state that the overlay represents. Additionally, overlays are used inconsistently to convey information about the state of a file.

## BRIEF SUMMARY

**[0008]** Features described herein are directed to a method for displaying an icon representing an underlying file on a user interface. The method may include providing a user with a size selection mechanism for selecting an icon size from a range of sizes. The method may further include searching a set of stored resources related to the underlying file and generating the icon in a user selected size based upon the stored resources.

**[0009]** In another aspect, a method is provided for enhancing display of icons on a user interface. The method may provide a user selection component for allowing a user to select an icon size for an icon representing a file. The method may additionally include selecting an icon type including one of a traditional icon and a thumbnail icon based on at least one of the size selection and the represented file. The method may additionally include generating the selected icon having the selected size and icon type.

**[0010]** In yet an additional aspect, a system is provided for enhancing icon display. The system may include a user icon scaling module for allowing a user to select an icon size for an icon representing an underlying file. The system may additionally include a resource selection module for selecting an image resource for generating the icon based on the size selection. The resource selection module may, in some aspects, proceed through three levels of potential icons: one having an image representing the contents of the displayed object, one having an image that displays details of the current object, and one having a generic image common to multiple objects.

**[0011]** In yet a further aspect, a system is provided for enhancing icon display. The system may include a user icon scaling module for allowing a user to select an icon size for an icon representing an underlying file. The system may additionally include a type selection module for selecting a type of icon including at least one of a traditional icon and a thumbnail icon, the type selection based on at least one of the size selection and the underlying file.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0012]** The features described in detail below are described with reference to the attached drawings figures, wherein:

**[0013]** FIG. **1** is a block diagram illustrating a computerized environment in which features described herein may be implemented;

**[0014]** FIG. **2** is a block diagram illustrating an icon scaling system in accordance with several features described herein;

**[0015]** FIGS. **3**A-**3**G are samples or different types and sizes of icons;

**[0016]** FIG. **4** is a block diagram illustrating an icon configuration;

**[0017]** FIG. **5** illustrates several thumbnail icon configurations;

**[0018]** FIGS. **6**A-**6**C illustrate multiple embodiments of visual effects for icons;

**[0019]** FIG. **7** illustrates a user scale for allowing a user to scale icons;

**[0020]** FIG. **8** is a flow chart illustrating a process for determining what type of icon to use;

**[0021]** FIG. **9** is a flow chart illustrating a process for accessing image resources; and

**[0022]** FIG. **10** is a flow chart illustrating an example process by which an icon may be selected for representing objects in a display.

**[0023]** FIG. **11** is a table showing example icon size relationships.

## DETAILED DESCRIPTION

### I. System Overview

**[0024]** Features described herein include a method and system for enabling scaling of icons to enhance a user experience. Icon scaling enables users to resize the icons in file management systems, such as Microsoft Windows Explorer, to any size in a range from a small size appropriate for looking at a long list of items to a very large size appropriate for looking at a small number of thumbnails within a single view. As an icon is resized, the appearance and layout of the icon may change to best accommodate its new size.

**[0025]** FIG. **2** illustrates an icon scaling module **200**. The icon scaling module **200** may be implemented as one or more program modules in a computerized environment as described below with reference to FIG. **1**. The icon scaling module **200** may include a resource selection module **210**, an overlay selection module **220**, a type selection module **230**, a label placement module **240**, and a user scaling module **250**. Through the user scaling module **250**, a user is able to select an icon size. Based on the user scaling selection, the type selection module **230** may select a type of icon such as a thumbnail or traditional icon. The resource selection module **210** may also select a particular image resource for scaling based on the scaling selection. Similarly, in some aspects, the overlay selection module **220** may select, scale, and/or position a status overlay and/or a type overlay based on the scaling selection, and the label placement module **240** may determine an appropriate location for a label associated with the icon based on the scaling selection.

**[0026]** Icons may be authored using a predetermined set of sizes, such as 16×16 pixels and 32×32 pixels. When a user chooses to view an icon at a size that is not one of the predetermined sizes, the scaling module **200** may interpolate to determine what image to show the user. When the size of an icon is changed, variables including the type of icon, the image resource, label placement, and/or overlay configuration may be affected as set forth herein.

### II. Exemplary Operating Environment

**[0027]** FIG. **1** illustrates an example of a suitable computing system environment **100** on which the features described herein may be implemented. The computing system environment **100** is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the features described herein. Neither should the computing environment **100** be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment **100**.

**[0028]** The features herein are described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The features may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, mainframe computers, and the like. The features may also be practiced in distributed computing environments, where tasks may be performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices.

**[0029]** With reference to FIG. **1**, the exemplary system **100** may include a general purpose-computing device in the form of a computer **110**, which may include a processing unit **120**, a system memory **130**, and a system bus **121** that couples various system components including the system memory to the processing unit **120**.

**[0030]** Computer **110** may include a variety of computer readable media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. The system memory **130** includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) **131** and random access memory (RAM) **132**. A basic input/output system **133** (BIOS), containing the basic routines that help to transfer information between elements within computer **110**, such as during start-up, is typically stored in ROM **131**. RAM **132** typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit **120**. By way of example, and not limitation, FIG. **1** illustrates operating system **134**, application programs **135**, other program modules **136**, and program data **137**.

**[0031]** The computer **110** may also include other removable/nonremovable, volatile/nonvolatile computer storage media. By way of example only, FIG. **1** illustrates a hard disk drive **141** that reads from or writes to nonremovable, nonvolatile magnetic media, a magnetic disk drive **151** that reads from or writes to a removable, nonvolatile magnetic disk **152**, and an optical disk drive **155** that reads from or writes to a removable, nonvolatile optical disk **156** such as a CD ROM or other optical media. Other removable/nonremovable, volatile/nonvolatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive **141** may be connected to the system bus **121** through a non-removable memory interface such as interface **140**, and magnetic disk drive **151** and optical disk drive **155** may be connected to the system bus **121** by a removable memory interface, such as interface **150**.

**[0032]** The drives and their associated computer storage media discussed above and illustrated in FIG. **1**, provide storage of computer readable instructions, data structures, program modules and other data for the computer **110**. In FIG. **1**, for example, hard disk drive **141** is illustrated as storing operating system **144**, application programs **145**, other program modules **146**, and program data **147**. Note that these components can either be the same as or different from operating system **134**, application programs **135**, other program modules **136**, and program data **137**. Operating system **144**, application programs **145**, other program modules **146**, and program data **147** are given different numbers here to illustrate that, at a minimum, they are different copies. A user may enter commands and information into the computer **110** through input devices such as a keyboard **162** and pointing device **161**, commonly referred to as a mouse, trackball or touch pad. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices may be connected to the processing unit **120** through a user input interface **160** that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor **191** or other type of display device may also be connected to the system bus **121** via an interface, such as a video interface **190**. In addition to the monitor, computers may also include other peripheral output devices such as speakers **197** and printer **196**, which may be connected through an output peripheral interface **195**.

**[0033]** The computer **110** may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer **180**. The remote computer **180** may be a personal computer, and typically includes many or all of the elements described above relative to the computer **110**, although only a memory storage device **181** has been illustrated in FIG. **1**. The logical connections depicted in FIG. **1** include a local area network (LAN) **171** and a wide area network (WAN) **173**, but may also include other networks.

**[0034]** When used in a LAN networking environment, the computer **110** may be connected to the LAN **171** through a network interface or adapter **170**. When used in a WAN networking environment, the computer **110** may include a modem **172** or other means for establishing communications over the WAN **173**, such as the Internet. The modem **172**, which may be internal or external, may be connected to the system bus **121** via the user input interface **160**, or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer **110**, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, FIG. **1** illustrates remote application programs **185** as residing on memory device **181**. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

**[0035]** Although many other internal components of the computer **110** are not shown, those of ordinary skill in the art will appreciate that such components and the interconnection are well known.

## III. Sample Icons

**[0036]** FIGS. **3**A-**3**G illustrate a sampling of icons in various sizes. FIG. **3**A illustrates a sampling of document icons **302**, **304**, **306**, **308**, **310**, **312**, **314**, **316**, and **318**. The document icons **302**, **304**, **306**, **308**, **310**, and **312** each include respective document images **302***a*, **304***a*, **306***a*, **308***a*, **310***a*, and **312***a*. Each of the aforementioned document icons additionally includes a type overlay **302***b*, **304***b*, **306***b*, **308***b*, **310***b*, and **312***b*. These type overlays indicate a type of the underlying document (e.g., word processing file, picture file, video file, audio file, spreadsheet, etc.). The document icon **302** is shown as having a 256×256 pixel overall icon size and a 60×60 pixel type overlay portion. The images and overlays may be scaled independently of one another, such that the image **304***a* is scaled between the images **302***a* and **306***a* and the overlay **304***b* is scaled between the overlays **302***b* and **306***b*. In the depicted example, document icon **306** is the next available document size. The overall size of icon **306** is 128×128 pixels and the overlay portion is 32×32 pixels. Document icon **304** is an icon scaled between the standard sized icons **302** and **306**. Similarly, document icon **308** is scaled between the document icon **306** and the document icon **310**, which includes a 64×64 overall icon size and an 18×18 pixel overlay portion **310***b*. Document icon **314** is a 32×32 pixel icon that includes only an overlay portion because its size is too small to clearly display both an overlay and a document image. Document icon **312**, which is scaled between document icon **310** and **314** show both the document image portion **312***a* and the type overlay portion **312***b*. Smaller icons **314**, **316** and **318** show only the type overlay portion of the item. The respective type overlays may include an icon, image, symbol, letter, number, etc. to indicate the type of document such as a MICROSOFT WORD™ document. The document icons may be provided in many sizes, including 16×16, 24×24, 32×32, 48×48, 128×128, and 256×256 pixels. Alternatively, some of these sizes may be omitted and/or additional sizes may be included.

**[0037]** FIG. **3**B illustrates a set of MICROSOFT POWERPOINT™ Icons **320**, **322**, **324**, **326**, and **328**. The Icons **320**, **322**, and **324** include an image portion **320***a*, **322***a*, and **324***a*, which may be an image of the first slide in a presentation. The icons **320**, **322**, and **324** also include overlay portions **320***b*, **322***b*, and **324***b*. The icons **326** and **328** are too small to be displayed with both portions. Accordingly these icons include only a thumbnail image. The decision to show a thumbnail image or an overlay may be made on a per type basis.

**[0038]** FIG. **3**C also includes a set of icons **330**, **332**, **334**, **336**, and **338**. The icons **330**, **332**, and **334** include respective image portions **330***a*, **332***a*, and **334***a*, which may be an image of a first page of a document, and overlay portions **330***b*, **332***b*, and **334***b* indicating that the document is an .mht document. Icons **336** and **338** are too small to adequately display both portions, and therefore include only the type overlay portion to indicate the type of underlying document.

**[0039]** FIG. **3**D illustrates a plurality of scaled photo icons **340**, **342**, **344**, **346**, and **348**. These icons are preferably thumbnail icons that include a photograph or image from within the underlying file. The thumbnail icons shown display useful information and include no overlay portion.

**[0040]** FIG. **3**E illustrates a set of video icons **350**, **352**, **354**, **356**, and **358**. The icons **350**, **352**, and **354** each include a respective image portion **350***a*, **352***a*, and **354***a*. The included images may be thumbnail images, such as a still image or portion thereof of a video. The icons **350**, **352**, and **354** each also include a respective type overlay portion **350***b*, **352***b*, and **354***b*. The icons **356** and **358** include only the image portion, which may be a thumbnail image, as the icons are too small to include both portions.

**[0041]** FIG. **3**F illustrates a set of contact icons **360**, **362**, **364**, **366**, **368**, and **370**, which may be used in conjunction with an email and/or address book. The icons **360** and **362** each include a respective image portion **360***b* and **362***b*, which may be a thumbnail image, and a respective informational portion **360***a* and **362***a*. Icons **364**, **366**, and **368** include only an image portion, as the icons are too small to show both the image and informational portion. Icon **370** illustrates a generic image that may be shown if a thumbnail image is unavailable.

**[0042]** FIG. **3**G illustrates a generic contact icon **372** that may be used when no photograph is available for the contact. The icon **372** includes a generic image **372***b* and an informational portion **372***a*.

**[0043]** As illustrated by these examples, scaled icons may differ in terms of their relative size. The scaling may be done beforehand, in which case multiple versions of the icon, at different scales, may be generated in advance and stored in a system memory. Scaling may also be performed dynamically, in which a new scale version of an icon is generated at the time it is needed. The different scale versions of icons may differ in other respects as well. For example, larger scale icons may have more graphical space available for additional elements, such as additional overlays, user interface elements, more detailed images, etc., and may be more complex than the smaller scale versions by including those additional features. Conversely, a smaller scale icon might omit some of these features, and may be simpler by comparison.

## IV. Overlays

**[0044]** As set forth above, the different icon sizes may also include different overlay sizes. As set forth above with respect to FIG. **2**, the selection and placement of overlays may be based on user-selected scaling. The overlay selection module **220** determines an overlay configuration based on user scaling input.

**[0045]** Two kinds of overlays include status overlays and type overlays. The overlays shown above with respect to FIGS. **3**A-G are type overlays that indicate a type. The type overlay may be used for thumbnail icons to convey what type of object the thumbnail image portrays. The type overlays appear in a lower right-hand corner of the image. Status overlays serve as a visual cue to help a user quickly identify some piece of relevant information about an item being viewed. Status overlays may appear in a lower left hand corner of an image.

**[0046]** One use of a status overlay includes alerting the user of a transitional or temporary state of a file upon which a user may wish to act. An example would be putting a “new” overlay on a file that has just recently been created. Another possible use for a status overlay is indicating relevant context sensitive information upon which a user may wish to act. An example would be putting an “error” overlay on a file in the download manager that could not complete downloading. In general, information that is being conveyed through a status overlay should also be a property that can be used for any purpose by the user, such as sorting, grouping, stacking, filtering, and searching. The information can also help the user make an informed decision as to what they want to/can do with a file. For example, an overlay might indicate that the file was downloaded from the Internet and has not been checked for viruses. Such an overlay would help the user decide whether to open the file, or whether to first run a virus scan on it. As another example, an overlay might indicate whether the associated file is available locally or if it is available remotely (such as from an Internet site). A file that is not available locally might take longer to open, and the user might factor that in the decision as to whether to open the file.

**[0047]** Table 1 describes size overlays that may be displayed on icons of different sizes.

TABLE 1

| Icon Size | Corresponding Overlay Size |
|---|---|
| 256 × 256 | 48 × 48 |
| 128 × 128 | 32 × 32 |
| 48 × 48 | 24 × 24 |
| 32 × 32 | 16 × 16 |
| 16 × 16 | 8 × 8 |

**[0048]** The overlays may scale independently from the icons. Between each defined icon snap size such as 256×256, 128×128, 64×64, 32×32, and 16×16, the overlays may scale linearly between the two overlay sizes. Table 2, shown in FIG. **11**, shows one embodiment of the overlay sizes relative to icon sizes along a continuous range.

## V. Icon Types And Layouts

**[0049]** FIG. **4** illustrates a traditional icon **400** including an icon image **401** and a status overlay **402**. The icon image **401** may be stored as a bitmap or may be dynamically generated. The status overlay **402** is used to convey status information about an item. The status overlay may also be dynamically generated or may be stored as a bitmap. In the displayed embodiment, the icon image **401** has dimensions of width=height=X. In some aspects, the status overlay **402** is drawn at the bottom left corner of the image so that the bottom edge of the overlay **402** is aligned with the bottom edge of the icon **400** and the left edge of the overlay **402** is aligned with the left edge of the icon **400**. Alternatively, the layout may be configured differently.

**[0050]** FIG. **5** illustrates possible layouts for a thumbnail icon. A row **500** illustrates bottom aligned thumbnail icons **510** and **520**. A row **530** illustrates centrally aligned thumbnail icons **540** and **550**. The bottom-aligned thumbnail icon **510** includes an image portion **512**, a status overlay **514**, and a type overlay **516**. The bottom-aligned thumbnail icon **520** includes an image portion **522**, status overlay **524**, and a type overlay **526**. The thumbnail icon **540** includes an image portion **542**, a status overlay **544**, and a type overlay **546**. The thumbnail icon **550** includes an image portion **552**, a status overlay **554**, and a type overlay **556**. In general, the image within each icon is denoted by (A), the status overlay by (B), and the type overlay by (C). The thumbnail icon itself is a square with a width and height equal to X. The thumbnail image (A) is scaled so that its longest dimension is at most equal to X.

**[0051]** As illustrated in FIG. **5**, in all of the displayed configurations, the thumbnail image (A) is drawn horizontally centered within the thumbnail icon. The vertical placement of the thumbnail image (A) may be determined by the type of view. If a current view has text labels placed below the icon then the bottom edge of the thumbnail image (A) may lie along the bottom edge of the thumbnail icon. If the current view has text labels placed to the side of the icon, then the thumbnail image (A) may be vertically centered within the thumbnail icon.

**[0052]** The placement of the status overlay (B) may depend on the placement of the thumbnail image (A) within the thumbnail icon. The depicted status overlay (B) fits within the boundaries of the thumbnail icon. Horizontally, the status overlay (B) may be centered around the left edge of the thumbnail image (A), but may be moved to the right when necessary, to avoid extending over a left boundary of the thumbnail icon itself.

**[0053]** The placement of the type overlay (C) may depend on the placement of the thumbnail image (A) within the thumbnail icon. Vertically, like the status overlay (B), the bottom edge of the type overlay (C) may lie along the bottom edge of the thumbnail image such that the type overlay (C) fits within the boundaries of the thumbnail icon when the thumbnail image (A) is bottom-aligned. Horizontally, like a mirror of the status overlay (B), the type overlay (C) may be centered around the right edge of the thumbnail image (A), but may be moved to the left when necessary, to avoid extending over the right boundary of the thumbnail icon itself.

**[0054]** For certain predefined icon sizes, such as icons between 16×16 and 47×47 pixels, the label placement module **240** may position the filename beside the icon. For example, for sizes 32×32 to 47, the label placement module **240** may provide additional tile information beside the icon. For sizes 48×48 and larger, the label placement module **240** may place the filename below the icon.

**[0055]** In a default mode, type overlays may be shown on a thumbnail icon if and only if it is within the document library. In some instances, type overlays will not be shown on photographs or video by default. A setting in the view options may be available to enable users to turn on or off the type overlays on documents. A mechanism such as an application program interface (API) may be provided that allows applications to associate a type overlay with each file type they own.

## VI. Visual Enhancements

**[0056]** FIGS. **6**A-**6**C illustrate visual enhancements that may be applied to thumbnail icons. There are several types of visual treatments that can be applied to thumbnail images and some of these treatments may further scale down an image.

**[0057]** FIG. **6**A illustrates a thumbnail image **600** and a type overlay **602**. A border **604** is applied and the icon may be registered to receive such a treatment. For example, MICROSOFT WORD™ documents may be registered to have this treatment applied to them. The border may be applied solely to the thumbnail image **600** and not to the entire bounding box of the thumbnail icon.

**[0058]** FIG. **6**B illustrates a thumbnail image **610** having a photo paper border **612**. The photo paper border **612** may be applied to a thumbnail if the underlying file is an "image" type and the icon size meets requirements. A photo paper border may be applied to the thumbnail image if the thumbnail image is an image type and it is of a suitable predetermined size, such as 256, 128, 48 or 32 pixels. In this example, no photo paper border is applied at size 16.

**[0059]** Table 3 describes the photo paper border width at each thumbnail icon size in this example.

TABLE 3

| Icon Size | Photo Paper Border Width |
|---|---|
| 256 | 5 |
| 128 | 3 |
| 48 | 2 |
| 32 | 1 |
| 16 | No photo paper border. |

**[0060]** The width may be interpolated between the displayed sizes in a similar manner to the overlays shown in Table 2.

**[0061]** FIG. **6**C illustrates a thumbnail image **620** having sprockets **622** and a type overlay **624**. The sprockets may be applied to the thumbnail image **620**, which may be a still image from a movie, to convey that it is a movie type. The sprockets include two images placed along the top and bottom of the thumbnail image **620** to make it look like a piece of film strip. A pair of sprockets may be applied to the thumbnail image if it is a perceived video type, and if the thumbnail icon size is of a predetermined size, such as 256, 128, 48, or 32 pixels. Again, in this example, no sprockets are applied at size 16. Table 4 describes the sprocket size at each thumbnail icon size in this example.

TABLE 4

| Icon Size | Sprocket Height |
|---|---|
| 256 | 18 |
| 128 | 10 |
| 48 | 6 |
| 32 | 4 |
| 16 | No sprockets. |

**[0062]** The top sprocket may be placed so that its bottom edge aligns with the top edge of the thumbnail image, and the bottom sprocket may be placed so that its top edge aligns with the bottom edge of the thumbnail image. If the thumbnail image has a height large enough to make the thumbnail image with the sprockets higher than X pixels, then the two sprockets may be moved closer to one another. The top sprocket may be moved down and the bottom sprocket may be moved up until the thumbnail image with the applied sprockets is no taller than the height X.

**[0063]** Other effects may include a gradient effect that may be applied to thumbnail image to make it more attractive. Furthermore, a shine effect may be applied to the thumbnail image if the object type has been registered to receive such a treatment. For example, POWERPOINT™ types may be registered to have this treatment applied to them. A thumbnail icon may also include a drop shadow effect or other visual effect.

## VII. Using the Scaling Mechanism

**[0064]** As set forth above, and as illustrated in FIG. **2**, a user scaling module **250** receives user input, such as through a graphical user interface, for scaling. FIG. **7** illustrates a user scale for allowing a user to scale icons. Users can scale icons to certain predetermined sizes, such as between 16×16 and

256×256 pixels. An icon size slider **702** may be provided in a view pane. A view mode control **700** allows a user to switch between an icon mode and a details mode. A user can move the icon slider **702** to adjust the icon size. The icon size slider **702** may have snap points around sizes 16, 32, 48, 128, and 256. If the slide control indicates a size within a few pixels of one of the snap sizes, the icons will snap to that size. In some aspects, if the slider control is within four pixels in either direction of a snap size, the selected icons will be set at the snap size.

**[0065]** Additionally, a view menu may have an item labeled "Icon Size," with the following choices appearing off a submenu: Tiny, Small, Medium, Large, and Huge, or similar selections that correspond to icon sizes such as 16, 32, 48, 128, and 256. For example, the view menu may be a pull-down menu listing selectable options, and the listed menu options may be selected by clicking, right-clicking, etc.

**[0066]** As with others of the inputs described above, such inputs may be entered using a graphical element, such as a slider bar. Furthermore, users may be able to scale icons using a hardware input, such as a scroll wheel found on many mice. Moving the scroll wheel forward while holding down the CTRL key can increase the size of the icon and moving the scroller wheel backward while holding down the CTRL key can decrease the size of the icon.

**[0067]** In absence of user scaling, the icons may have default sizes. The following Table 5 describes example default sizes of icons in various folder locations.

TABLE 5

| Place | Default Icons Size | Default Details Size |
|---|---|---|
| Document Library | 48 | 16 |
| Photo/Video Library | 128 | 16 |
| Music/Radio Library | 48 | 16 |
| Contact Library | 48 | 48 |

**[0068]** If the user selection requires that the icon be scaled between snap sizes, the system utilizes a scaling mechanism including a mechanism for converting the icons to an appropriate format, such as a graphical texture provided by MICROSOFT DIRECTX(™) graphics technology. A hardware accelerated scaling mechanism may implement a bilinear filter or other known mechanism to perform the actual icon scaling while maintaining visual fidelity of the image. The scaling may occur in real-time by the icon scaling system **200**. The real-time scaling can occur at draw time or optionally at other times.

## VIII. Type Selection Process

**[0069]** FIG. **8** is a flow chart illustrating a process implemented by the type selection module **230**, shown in FIG. **2**, for determining which type of icon should be used after a user has selected a particular icon size as explained above with respect to FIG. **7**. As set forth above with regard to FIG. **2** the selection of a thumbnail icon or traditional icon typically depends upon the scaling selection of the user.

**[0070]** In step A**02**, the system compares the icon size and a pre-set size cutoff. If the icon size is above the pre-set size cutoff in step A**04**, the system seeks a thumbnail icon in step A**10**. If the icon is not above the size cutoff in step A**04**, the system selects and shows a traditional item in step A**14**.

**[0071]** In step A**10**, the system seeks a thumbnail icon. If a thumbnail icon is not cached in step A**12**, the system attempts to extract it in step A**06**. If the thumbnail is not cached in step A**12**, or if the extraction is successful in step A**08**, the system selects and shows the thumbnail icon in step A**16**. If the extraction is unsuccessful in step A**08**, the system selects and shows the traditional icon in step A**14**.

**[0072]** Exemplary icon size cutoffs are illustrated in Table 6 below.

TABLE 6

| Perceived Item Type | Thumbnail/Icon Cutoff |
|---|---|
| Document | 32 |
| Folder | 32 |
| Picture | Always thumbnail |
| Video | Always thumbnail |
| Stack (Phodeo) | Always thumbnail |
| Song | Always thumbnail |
| Stack (Album) | Always thumbnail |
| Stack (Artist) | Always thumbnail |
| People | Always thumbnail |
| Groups | Always thumbnail |
| Devices | Always thumbnail |
| Internet Shortcuts | 16 |

## IX. Resource Selection And Scaling Process

**[0073]** FIG. **9** illustrates a process for selection of an image resource by the resource selection module **210**, illustrated in FIG. **2**, after a user has scaled a selected icon.

**[0074]** As set forth above, the system may provide image snap sizes. The sizes might include for example 16, 32, 48, 128, and 256 pixels. For every item Y at any size X the system may choose the image at the best available size for Y. In step B**02**, the system searches for the appropriate sized image X. If X is a snap size and an image for Y is available at size X in step B**04**, then the system uses the available image at size X in step B**20**. If an image for Y is not available in size X, in step B**04**, then the system determines if the largest size was searched in step B**06**. If the largest size was not searched in step B**06**, the system searches the next largest size in step B**08**. If the next largest image is available in step B**10**, the system uses the available icon in step B**20**. If the image is not available in step B**10**, the system continues the search until the largest image has been searched.

**[0075]** Subsequently, in step B**12**, if the search for larger images representing Y was unsuccessful, the system determines if the smallest size has been searched. If the smallest size has not been searched, the system continues to search for the next smallest size in step B**14**. If the image is available in step B**16**, the system uses it in step B**20**. If the icon is not available and the smallest size has been searched in step B**12**, the system uses a generic image in step B**18**. If and only if no image can be found at any size, the system will show a generic image. Although the process above is described as considering larger images prior to smaller images, the reverse process is also encompassed.

**[0076]** If the user selection requires that the image be scaled between snap sizes, the system utilizes a scaling mechanism including a mechanism for converting the images to an appropriate format, such as DX9 textures. A hardware accelerated scaling mechanism may implement a bilinear filter to perform the actual image scaling. In addition to scaling the retrieved image, the system also attempts to re-extract at the correct size. The system uses the closest stored image it can find in the interim time. Once the system re-extracts, it displays the correctly sized image and also stores the new correctly sized image into a cache for subsequent retrieval. This extraction, display, store-into-cache sequence also occurs if the cache is empty and first displays the traditional icon.

**[0077]** The resource selection and/or scaling process may also use a variety of different types of icons depending on resource availability. For example, FIG. **10** depicts an example process in which the resource selection module **210** may select an appropriate icon for display. First, in step **1001**, the system may receive an icon scaling size to be used for the icons appearing in the display. This may be based on a user preference, such as the scaling mechanism described above, or it may be done automatically by the system as a default or user setting from hardware and/or software inputs.

**[0078]** Once the icon scaling size is determined, the system may then proceed to step **1002**, in which the next (or first) object to be displayed may be considered. The objects may be, for example, the various files, documents, data, etc. that are to be represented by the icons in the display. In step **1003**, the system may check to see whether an imaging icon, having a reduced-size image displaying the current object's contents, is available for the current object. This may be accomplished using, for example, the resource selection process described above and shown in FIG. **9**. Other approaches may also be used. For example, the system may make the determination based on whether a predefined imaging icon already exists for the current object. Alternatively, the system may make the determination based on the object type, or its data contents, to decide whether a suitable imaging icon may be generated for the display.

If, in step **1003**, the system determines that a suitable imaging icon is available, the process may move to step **1004**, in which that suitable imaging icon may be retrieved and/or generated, and used for the display of the current object. If, however, the system determines that no suitable imaging icon is available, the process may move to step **1005**.

In step **1005**, the system may check to see if a suitable detail icon is available for the object. A detail icon may be any displayable image whose appearance depends on the contents of the current object, but which falls short of an image of the object's contents. For example, the detail icon might not include a display showing the object's contents in visual form (e.g., a miniature image of a picture file), but it may include some indicia of a type (e.g., whether it is a word processing document, a text document, a spreadsheet, an audio file, a video file, etc.), status (e.g., whether it is editable, its age, when it was last edited, etc.), or other characteristic of the object. Although the detail icon does not have an image representation of the object's contents, the detail icon may still include helpful information identifying the current object. For example, some or all of the overlay information described above may be incorporated or added into the detail icon, such that the detail icon conveys information identifying one or more characteristics (e.g., object type, status, contents, etc.) of the object. For example, if the object is a word processing document (e.g., a MICROSOFT WORD™ document), but the system decided against displaying an image of the object's contents (perhaps the document text was too dense to be adequately represented as a thumbnail), the detail icon might display a symbol representing the fact that the object is a word processing document. The detail icon may include, for example, a type overlay identifying the document type, such as overlay **302***b*. Other object characteristics and/or overlay data may also be displayed in the detail icon.

The step **1005** determination may follow the same process used in the step **1003** determination, and may check to see if a suitable detail icon can be retrieved and/or generated for the current object. Alternatively, the system may be configured to avoid trying to generate detail icons, and may instead simply check to see if a predetermined detail icon (e.g., a preconfigured bitmap image) already exists in the system. Simplifying this step may lead to faster performance.

If, in step **1005**, the system determines that a suitable detail icon is available for the object, the system moves to step **1006**, and uses that suitable detail icon in the display for the current object. However, if no suitable detail icon is available, then the system may move to step **1007**.

In step **1007**, the system has determined that no suitable imaging or detail icons are available, and may use a generic icon to represent the current object in the display. A generic icon may be an image that identifies the existence of an object, but which provides minimal information regarding the file (or none at all). For example, a generic icon might be a plain shape, such as a square, or a blank sheet of paper, to represent the existence of an object. A generic icon may be a predefined image that is common to multiple objects, and in some instances, may have a constant appearance that does not vary with the object contents, type, status, etc. For example, a single generic icon may be defined for use with all objects of unknown file type to the system.

By step **1008**, the system will have determined whether to use an imaging icon, a detail icon, or a generic icon for the current object, and may then check to see if any other objects are to be displayed. For example, if the FIG. **10** process is occurring in order to display the contents of a particular project folder, the step **1008** check may determine whether it has identified the appropriate icon (e.g., imaging, detail, or generic) for all objects to be displayed. If additional objects remain to be processed, the system may return to step **1002** to retrieve the next object and loop through the steps described above to identify an appropriate icon for the next object.

If no further objects remain to be processed, then the system may terminate the resource selection process in step **1009**.

## X. Conclusion

**[0079]** Using one or more of the features described above, various icon images may be generated. A different type of icon image may be selected based on the location of the file represented by the icon (e.g., files in a particular folder on the system all share a common icon) and/or the characteristics of the file (e.g., its metadata, type, how recently it was accessed and/or edited etc.). Additionally, icon images may be selected using contents of the file (e.g., a shortcut icon might retrieve the contents of the underlying file to determine the icon to be displayed).

Thus, icon scaling enables users to resize icons to any size in a range from a small size to a very large size, depending upon user preference. As an icon is resized, the appearance of the icon and the layout of the icons changes to best accommodate its new size.

**[0080]** Users can, in real-time, scale icons across the continuous range of possible sizes, and the scaling is visually appealing. Thumbnail icons may consistently convey what type of object they represent (e.g. Word Document, PowerPoint deck, video clip, etc.) even for thumbnail extractors written by ISVs. Overlays may be used in a consistent and reliable way as visual cues conveying some information about an item.

**[0081]** While particular features have been illustrated and described in detail herein, it should be understood that various changes and modifications might be made without departing from the scope and intent of the description herein. The features described herein are intended in all respects to be illustrative rather than restrictive. Alternatives will become apparent to those skilled in the art. For example, the scaling processes described above may work sufficiently well when scaling up and scaling down the icon size. As another example, although the description above refers to icons for underlying files, the system is not limited to representing files, and can be used for icons representing any item on a computing system, such as directories, shortcuts, etc.

**[0082]** From the foregoing it will be seen that the features described herein well adapted to attain some or all of the ends and objects set forth above. It will also be understood that certain features and sub-combinations may be employed without reference to other features and sub-combinations. For example, the steps in the processes described above may be rearranged and/or omitted as desired.

What is claimed is:

**1**. A method performed by a computing system having a processor and memory for displaying an icon having a type overlay that upon selection allows access to an underlying file, the method comprising:

generating, with the processor and the memory, the icon for the underlying file based on a user-requested icon size received from a user, wherein the user-requested icon size is an icon size between a first defined icon size and a second defined icon size;

selecting a type overlay overlapping and obstructing a portion of the icon, wherein the type overlay indicates a computing application associated with the underlying file;

determining a size of the type overlay based on the user-requested icon size, wherein the size of the type overlay scales independently from the icon between the first defined icon size and the second defined icon size; and

presenting the icon and the type overlay overlapping and obstructing a portion of the icon.

**2**. The method of claim **1**, further comprising providing the user with a size selection user interface element having a continuous range of sizes.

**3**. The method of claim **2**, further comprising searching a set of stored image resources that includes a set of images having a number of defined sizes.

**4**. The method of claim **2**, further comprising interpolating between the first defined icon size and a second defined icon size.

**5**. The method of claim **4**, wherein the scaling applied to the icon is different from the scaling applied to the type overlay to interpolate between the first defined icon size and the second defined icon size.

**6**. The method of claim **1**, further comprising positioning the type overlay based upon the identified icon size.

**7**. The method of claim **6**, further comprising positioning the type overlay in a corner of the generated icon.

**8**. The method of claim **6**, further comprising positioning a status overlay in a corner of the generated icon.

**9**. The method of claim **1**, wherein the icon is generated as a detail icon based upon the user-requested size.

**10**. The method of claim **1**, wherein the icon is generated as a thumbnail image icon based upon the user-requested size.

**11**. The method of claim **1**, wherein the icon is generated as a generic image icon based upon the user-requested size.

**12**. A computer readable medium storing computer executable instructions that, when executed by a processor and memory, perform a method for displaying an icon having a type overlay that upon selection allows access to an underlying file, the method comprising:

providing a user-selection component for allowing a user to identify an icon size for an icon representing an underlying file;

receiving a user-requested icon size from the user, wherein the user-requested icon size identifies an icon size for the icon representing the underlying file;

generating, with the processor and the memory, the icon for the underlying file based on a user-requested icon size received from a user, wherein the user-requested icon size is an icon size between a first defined icon size and a second defined icon size;

selecting a type overlay overlapping and obstructing a portion of the icon, wherein the type overlay indicates a computing application associated with the underlying file;

determining a size of the type overlay based on the user-requested icon size, wherein the size of the type overlay scales independently from the icon between the first defined icon size and the second defined icon size; and

presenting the icon and the type overlay overlapping and obstructing a portion of the icon.

**13**. The method of claim **12**, wherein said selection component displays a continuous range of sizes for user selection.

**14**. The computer readable medium of claim **12**, wherein the method further comprising searching a set of stored image resources upon receiving the user-requested icon size, the stored set of image resources including a set of images in a plurality of defined sizes.

**15**. The computer readable medium of claim **14**, wherein the method further comprises interpolating between the first defined icon size and the second defined icon size to generate the icon.

**16**. The method of claim **12**, wherein the type overlay overlaps and obstructs a bottom right corner of the icon.

**17**. The computer readable medium of claim **16**, further comprising positioning a status overlay in a bottom left corner of the generated icon.

**18**. A computer readable medium storing computer executable instructions that, when executed by a processor and memory, perform a method for displaying an icon having a type overlay that upon selection allows access to an underlying file, the method comprising:

providing a user-selection component for allowing a user to identify an icon size for an icon representing an underlying file, wherein the icon size is between a first defined icon size and a second defined icon size;

receiving a user-requested icon size from the user, wherein the user-requested icon size identifies an icon size for the icon representing the underlying file such that the user-requested icon size is between the first defined icon size and the second defined icon size;

generating, with the processor and the memory, the icon for the underlying file based on a user-requested icon size received from a user, wherein the icon is generated utilizing a first ratio between the first defined icon size and the second defined icon size;

selecting a type overlay overlapping and obstructing a portion of the icon, wherein the type overlay indicates a computing application associated with the underlying file;

determining a size of the type overlay based on the user-requested icon size, wherein the type overlay is generated utilizing a second ratio between the first defined icon size and the second defined icon size, such that the first ratio and the second ratio are different ratios; and

presenting the icon and the type overlay overlapping and obstructing a portion of the icon.

**19**. The method of claim **18**, wherein the first ratio is between 1:1 (256/256) and 1:2 (128/256), and the second ratio is between 1:1 (48/48) and 2:3 (32/48).

**20**. The method of claim **18**, wherein the first ratio is between 1:1 (128/128) and 3:8 (48/128), and the second ratio is between 1:1 (32/32) and 3:4 (24/32).

* * * * *