

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
6 July 2006 (06.07.2006)

PCT

(10) International Publication Number
WO 2006/072049 A2

(51) International Patent Classification:

G06F 17/24 (2006.01) **G06F 3/00** (2006.01)
G06F 17/21 (2006.01)

(21) International Application Number:

PCT/US2005/047575

(22) International Filing Date:

30 December 2005 (30.12.2005)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

11/027,577 30 December 2004 (30.12.2004) US

(71) Applicant (for all designated States except US): **SAP, AKTIENGESELLSCHAFT** [DE/DE]; Neurottstrasse 16, 69190 Waldorf (DE).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **BROOKLER, David** [US/US]; 1875 Century Park East #H-2538, Century City, CA 90067 (US). **WEINBERG, Paul**; 2160 Century Park East #1950, Century City, CA 90067 (US).

(74) Agent: **MAYO, Joseph, J.**; Dalina Law Group PC, 7910 Ivanhoe Ave #325, La Jolla, CA 92037 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

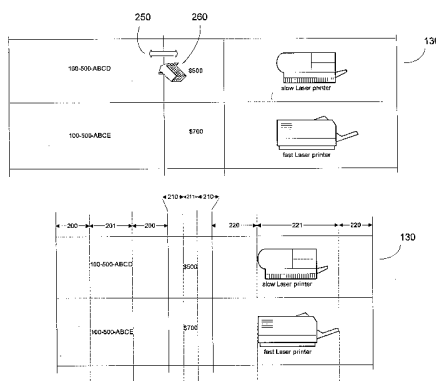
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, 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:

— without international search report and to be republished upon receipt of that report

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.

(54) Title: METHOD FOR REALLOCATING TABLE FORMATS IN REAL-TIME



(57) Abstract: Embodiments of the invention improve upon existing auto formatting functions for tables presented in a graphical user interface by providing an improved mechanism for changing table columns in real-time. One embodiment of the invention incorporates a callback or module in a program implementing the mechanism for changing table columns in real-time that intercepts a table column resize event and automatically resizes the other columns in the table. In resizing the table columns the module can employ various approaches. For instance, during a resize event the module may calculate the amount of space around each word in each column and readjust the amount of space around the text in order to keep the columns a size that has substantially equivalent white space around each of the respective column entries. Another strategy the module may employ is to adjust the columns to comprise an approximately equivalent width ratio as the table is made wider or narrower. The system may utilize these and other resizing approaches to arrive at a desired result. In addition, columns may be designated as frozen in which case they are not altered while the other columns are altered. Columns may alternatively be set to unfrozen to allow the columns to resize along with other columns using the methods of one or more embodiments of the invention.

METHOD FOR REALLOCATING TABLE FORMATS IN REAL-TIME

**INVENTORS:
DAVID BROOKLER
PAUL WEINBERG**

BACKGROUND OF THE INVENTION**FIELD OF THE INVENTION**

[001] Embodiments of the invention described herein pertain to the field of graphical user interfaces. More particularly, but not by way of limitation, one or more embodiments of the invention enable systems to intelligently format table components within a graphical user in real-time.

DESCRIPTION OF THE RELATED ART

[002] The problem with altering column sizes in presentation layer tables (for example in a graphical page layout program for a catalog) is that when a user readjusts a column width or inserts a new column, the already existing columns do not proportionally readjust while maintaining the general table structure and overall total table width. Because the column widths do not change in proportion, users are required in various instances to engage in the laborious process of altering each column by hand in order to achieve the proper proportions, each time they want to adjust the width of a single column. Without manual readjustment the columns often run off the page and/or the text within the columns is arranged in a way that is visually displeasing. Existing "auto format" functions are inadequate in that they fail to properly reallocate the format of the table in a visual pleasing way. Hence there is a need for software configured to properly reallocate table formats in real time.

BRIEF SUMMARY OF THE INVENTION

[003] Embodiments of the invention improve upon existing auto formatting functions for tables presented in a graphical user interface by providing an improved mechanism for changing table columns in real-time. One embodiment of the invention incorporates a callback or module in a program implementing the mechanism for changing table columns in real-time that intercepts a table column event for example a resize event or text event and automatically resizes the other columns in the table in addition to the column being altered.

[004] In resizing the table columns the module can employ various approaches employing a design pattern such as a strategy pattern than may be switched during operation. For instance, during a resize event the module may calculate the amount of space around each word or object in each column and readjust the amount of space around the text or object in order to keep the columns a size that has substantially equivalent white space around each of the respective column entries. Alternatively, or during the alteration of a column a user may switch the operation that keeps the ratio of the sizes of the columns the same regardless of the white space in each column and the operation may be switched while the altering a column so that alternative strategies for column widths may be viewed rapidly until the desired look is achieved. Table column adjustments may change the overall table width or be set to keep the overall table width constant. In addition, columns may be designated as frozen in which case they are not altered while the other columns are altered. Columns may alternatively be set to unfrozen to allow the columns to resize along with other columns using the methods of one or more embodiments of the invention.

[005] Another strategy the module may employ is to adjust the columns to comprise an approximately equivalent width ratio as the table is made wider or narrower. A combination of white space and overall column ratios may also be utilized as when a function is utilized to combine these approaches, for example the average between the white space operation and the ratio operation. The system may utilize these and other resizing approaches to arrive at a desired result.

[006] Another strategy the module may employ is to adjust all columns of all tables comprising a given element. Embodiments of the invention may be configured

to automatically adjust all columns in all tables in a document comprising the element when for example a database field is updated or when the user changes an element in a column that exists in a plurality of tables. Changing an element may comprise typing into, deleting text from, pasting an object into, or deleting an object from a column. Altering a column may comprise changing an element or altering the column width itself, for example by dragging the column divider with a mouse to increase or decrease the size of the column itself.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows a computer display coupled with a keyboard and mouse and shows a graphical user interface table displaying multiple columns and rows.

Figure 2A shows a table with a column in the process of being resized by dragging a mouse to the left or right.

Figure 2B shows the resulting table columns sized based on the white space surrounding the elements in the columns.

Figure 3A shows a table with a column in the process of being resized by dragging a mouse to the left or right.

Figure 3B shows the resulting table columns sized based on the sizes of the elements themselves residing in the columns.

Figure 4A shows a table with a column in the process of being resized by dragging a mouse to the left or right.

Figure 4B shows the resulting table columns sized to comprise the ratios of the columns in Figure 4A.

Figure 5A shows a table with a column in the process of being resized by dragging a mouse to the left or right and dragging the mouse up or down or using a scroll wheel on the mouse.

Figure 5B shows the resulting table columns sized to comprise the ratios of the columns in Figure 5A as shown by the selected “%” strategy selected in the popup in Figure 5A.

Figure 6A shows a table with a column in the process of being resized by dragging a mouse to the left or right and dragging the mouse up or down or using a scroll wheel on the mouse.

Figure 6B shows the resulting table columns sized based on each item in the table shown in Figure 6A as shown by the selected “ITEM” strategy selected in the popup in Figure 6A.

Figure 7A shows a table with a column in the process of being updated via a database update to a field and via dragging the mouse up or down or using a scroll wheel on the mouse to select the strategy employed in updating the column sizes.

Figure 7B shows the resulting table columns sized based on each item in the table shown in Figure 7A as shown by the selected “ITEM” strategy selected in the popup in Figure 7A.

Figure 8A shows a table with a column in the process of being updated via a database update to a field and via dragging the mouse up or down or using a scroll wheel on the mouse to select the strategy employed in updating the column sizes.

Figure 8B shows the resulting table columns sized based on each item in the table shown in Figure 8A as shown by the selected "ITEM" strategy selected in the popup in Figure 8A and also shows a second table being automatically updated when it comprises the same database field.

Figure 9A shows a table with a column to be resized by globally replacing text in at least one cell of the table.

Figure 9B shows the resulting table columns sized to comprise the ratios of the columns in Figure 9A.

Figure 10 shows a flow chart of the process of reallocating table formats in real-time.

DETAILED DESCRIPTION OF THE INVENTION

[007] Embodiments of the invention improve upon existing auto formatting functions for tables presented in a graphical user interface by providing an improved mechanism for changing table columns in real-time. One embodiment of the invention incorporates a callback or module in a program implementing the mechanism for changing table columns in real-time that intercepts a table column event for example a resize event or text event and automatically resizes the other columns in the table in addition to the column being altered.

[008] In the following exemplary description numerous specific details are set forth in order to provide a more thorough understanding of embodiments of the invention. It will be apparent, however, to an artisan of ordinary skill that the present invention may be practiced without incorporating all aspects of the specific details described herein. Any mathematical references made herein are approximations that can in some instances be varied to any degree that enables the invention to accomplish the function for which it is designed. In other instances, specific features, quantities, or measurements well-known to those of ordinary skill in the art have not been described in detail so as not to obscure the invention. Readers should note that although examples of the invention are set forth herein, the claims, and the full scope of any equivalents, are what define the metes and bounds of the invention.

[009] In resizing the table columns the module can employ various approaches employing a design pattern such as a strategy pattern than may be switched during operation. For instance, during a resize event the module may calculate the amount of space around each word or object in each column and readjust the amount of space around the text or object in order to keep the columns a size that has substantially equivalent white space around each of the respective column entries. Alternatively, or during the alteration of a column a user may switch the operation that keeps the ratio of the sizes of the columns the same regardless of the white space in each column and the operation may be switched while the altering a column so that alternative strategies for column widths may be viewed rapidly until the desired look is achieved. Table column adjustments may change the overall table width or be set to keep the overall table width constant. In addition, columns may be designated as frozen in which case they are not altered while the other columns are altered. Columns may

alternatively be set to unfrozen to allow the columns to resize along with other columns using the methods of one or more embodiments of the invention.

[0010] Another strategy the module may employ is to adjust the columns to comprise an approximately equivalent width ratio as the table is made wider or narrower. A combination of white space and overall column ratios may also be utilized as when a function is utilized to combine these approaches, for example the average between the white space operation and the ratio operation. The system may utilize these and other resizing approaches to arrive at a desired result.

[0011] Another strategy the module may employ is to adjust all columns of all tables comprising a given element. Embodiments of the invention may be configured to automatically adjust all columns in all tables in a document comprising the element when for example a database field is updated or when the user changes an element in a column that exists in a plurality of tables. Changing an element may comprise typing into, deleting text from, pasting an object into, or deleting an object from a column. Altering a column may comprise changing an element or altering the column width itself, for example by dragging the column divider with a mouse to increase or decrease the size of the column itself.

[0012] Figure 1 shows computer display 100 coupled with keyboard 110 and mouse 120 and shows a graphical user interface table 130 displaying multiple columns and rows. Table 130 comprises three columns with product number, price and product image headings. The contents of table 130 are exemplary and one skilled in the art will recognize that the contents of the columns may vary or comprise different arrangements and contents and may comprise more or less columns or rows. Computer display 100 may comprise an internal central processing unit or may be coupled optionally to an external computer packaged in a variety of formats.

[0013] Figure 2A shows a subset of table 130 with a column in the process of being resized by dragging mouse 260 to the left or right as shown by arrow 250. Figure 2B shows the resulting table columns sized based on the white space surrounding the elements in the columns. Utilizing an equal white space strategy dimensions 200, 210 and 220 can be kept equal to one another while the first column

in this example is being resized. Alternatively, white space dimension 200 may differ from the white space dimension 210 and 220 if a strategy of utilizing white space in proportion to each column width is utilized or if the white space on each side of each column is designated as constant. In the later cases, white space dimension 200 may be larger than white space dimension 210 and smaller than white space dimension 220 but is not shown for brevity. In all of these cases leftmost dimension 200 is equal to rightmost dimension 200, leftmost dimension 210 is equal to rightmost dimension 210 and leftmost dimension 220 is equal to rightmost dimension 220. Note that the printer in the first row comprises a picture that is narrower than the printer in the second row and that the printer in the first row is left-justified. This not required and the printer in the first row may be center-justified or right-justified or zoomed in on so that it fills the space of dimension 221. As shown, overall table width of Figure 2B is shown as altered with respect to the table width of Figure 2A, however the width of the table may remain a constant depending on the mode of the alteration as set by the strategy pattern in use for the table. Keeping the overall table width the same while spreading equal white space among the columns is shown as subset of table 130 in Figure 2C is adjusted using this strategy with the resulting columns sized with equal white space in Figure 2D. In addition, by setting a given column to “frozen”, that column is not altered in width while the other columns are adjusting. For example, if the middle column of Figure 2A is set to frozen then dimension 210 of the middle column of Figure 2B is not altered while mouse 260 moves left or right as shown by arrow 250. Dimensions 200 and 220 then are utilized in the readjustment method and in an equal white space strategy are kept equal to one another while adjusting wherein dimension 210 remains constant. Keeping the overall table width the same while spreading equal white space over the unfrozen columns results in two dimensions therefore not locked in this strategy which are dimension 201 and dimension 221. Dimensions 201 and 221 may therefore zoom in or zoom out with respect to the corresponding contents of the associated columns. For the case of a ratio based strategy, for example if the rightmost column in Figure 2A is 60% larger than the leftmost column in Figure 2A, then as mouse 260 is dragged left or right, the white space of dimension 220 may grow by 60% more than dimension 200 in the variable size table strategy and in the fixed size table strategy, would change by the same amount although the entities in the columns would zoom to account for the changes required to dimensions 201 and 221 to keep the overall table width the same. A column may be set to frozen or

unfrozen through any available user gesture including right clicking on a column and asserted a check box for example.

[0014] Note that although for brevity keeping the table widths the same or adjusting a column without maintaining constant table width is assumed in the following figures and as one skilled in the art will recognize is a mode of operation in keeping with the spirit of the present invention. These table width modes are referred to as VARIABLE or CONSTANT table width modes. In addition, freezing columns and unfreezing columns to allow the column to remain unaltered or to be altered respectively is assumed as another mode of operation in keeping with the spirit of the invention in all strategies listed herein and allows columns to ignore or take part in the column alteration operation at hand. Furthermore, adjusting a column may comprise a scope related to the current table or to the entire document. For example, adjusting a column comprising a given piece of data may allow the user to adjust all tables comprising the data at once if the scope of the change is set to document scope, otherwise the alteration is applied at the scope of the table being altered.

[0015] Figure 3A shows a subset of table 130 with a column in the process of being resized by dragging mouse 260 to the left or right as shown by arrow 250. Figure 3B shows the resulting table columns sized based on the sizes of the elements themselves residing in the columns. Utilizing an item based strategy dimensions 200, 210 and 220 can be kept equal to a fraction of the size of the element in each column. In this example, dimension 200 is set to 66% of dimension 201, dimension 210 is set to 100% of dimension 211 and dimension 220 is set to 50% of dimension 221 while the first column in this example is being resized. Alternatively, white space dimensions, 200, 210 and 220 may be set to the same percentage, for example 50% of the respective element dimensions 201, 211, and 221. The overall table width using this strategy may be held constant by increasing the size of the column elements. The table width mode may be set to CONSTANT or VARIABLE in order to allow the strategy pattern in use to maintain the overall table width or allow modification of a column to alter the overall table width. This is true for all strategy patterns.

[0016] Figure 4A shows a subset of table 130 with a column in the process of being resized by dragging mouse 260 to the left or right as shown by arrow 250.

Figure 4B shows the resulting table columns sized to comprise the ratios of the columns in Figure 4A. Utilizing a ratio based strategy dimensions 401, 411 and 421 can be kept equal to the ratio of the columns of the table as seen in Figure 4A. For example if the columns in subset of table 130 in Figure 4A measure 410, 160 and 460 pixels, then the column dimensions 401, 411 and 421 would be 308, 120 and 345 pixels respectively for a VARIABLE table width strategy mode adjustment. Using a CONSTANT table width mode, the strategy pattern will not simply decrease one table column and increase another table column for a table comprising a triality of columns but will rather spread the increase in one column size to the remaining plurality of columns. For example if column width 411 increases by 100 pixels with a user dragging mouse 260 to the left direction of arrow 250, then instead of simply leaving the rightmost column the same size and decreasing the size of the leftmost column by 100 pixels, the leftmost column can be decreased by 50 pixels and the rightmost column can be decreased by 50 pixels which would be termed an equal ratio strategy. Using a CONSTANT ratio strategy would involve decreasing column width 401 by 40 pixels and decreasing column width 421 by 60 pixels if column width 401 is for example 400 pixels and column width is 600 pixels before the adjustment. This would yield a constant 10% decrease in the leftmost and rightmost columns.

[0017] Figure 5A shows a subset of table 130 with a column in the process of being resized by dragging mouse 260 to the left or right as shown by arrow 250 and dragging the mouse up or down or using a scroll wheel on the mouse to select a strategy in strategy popup 500. Figure 5B shows the resulting table columns sized to comprise the ratios of the columns in Figure 5A as shown by the selected “%” strategy selected in the popup in Figure 5A. In this case the “%” or ratio strategy is selected resulting in a table resized with column dimension ratios set to remain the same without regard for resulting white space. Holding down the shift key for example during the move may be used to switch between VARIABLE and CONSTANT table width mode. Any other mechanism for setting the mode between VARIABLE and CONSTANT table width mode including setting a menu item or check box in a popup are in keeping with the spirit of the invention.

[0018] Figure 6A shows a subset of table 130 with a column in the process of being resized by dragging mouse 260 to the left or right as shown by arrow 250 and

dragging the mouse up or down or using a scroll wheel on the mouse to select a strategy in strategy popup 500. Figure 6B shows the resulting table columns sized based on each item in the table shown in Figure 6A as shown by the selected "ITEM" strategy selected in the popup in Figure 6A. In this case the "ITEM" or item strategy is selected resulting in a table resized with column dimensions resized as described with regards to Figs. 3A and 3B. Although shown as a VARIABLE mode adjustment, a CONSTANT table width mode adjustment is also possible by as per the CONSTANT mode discussion with relation to Figs. 3A and 3B.

[0019] Figure 7A shows a subset of table 130 with a column in the process of being updated via a database update to a field and via dragging the mouse up or down or using a scroll wheel on the mouse to select the strategy employed in updating the column sizes. Figure 7B shows the resulting table columns sized based on each item in the table shown in Figure 7A as shown by the selected "ITEM" strategy selected in the popup in Figure 7A. In this case the "ITEM" or item strategy is selected resulting in a table resized with column dimensions resized as described with regards to Figs. 3A and 3B. For the case where families of data are used in a hierarchical manner the strategy selected may be obtained from a parent table if the strategy has not been selected for the child table. United States Patent Application Serial Number 09/960,902 is hereby incorporated herein by reference. A parent family may comprise Laser Printers while the child families may comprise Acme Laser Printers and Best Laser Printers for example. In this case, if the graphical user interface table comprising Acme Laser Printers for example has no associated strategy then the Laser Printer family may be accessed in order to determine if a strategy for reallocating table formats has been selected for the parent family. There is no requirement that a family table has to be displayed in a graphical format in a catalog for example. The family may be used merely to set up the types of formatting associated with graphical representations of the families during presentation. CONSTANT and VARIABLE mode adjustments may be set on a per table or per document basis via any method such as a popup or menu item setting or control key for example.

[0020] Figure 8A shows a subset of table 130 with a column in the process of being updated via a database update to a field and via dragging the mouse up or down or using a scroll wheel on the mouse to select the strategy employed in updating the

column sizes. Figure 8B shows the resulting table columns sized based on each item in the table shown in Figure 8A as shown by the selected "ITEM" strategy selected in the popup in Figure 8A and also shows a second table being automatically updated when it comprises the same database field. This example shows that table 830 also comprises the database field as displayed in the printer column of the second row of the left table shown in Fig. 8B. In this example, table 830 is resized when subset of table 130 is resized after the user is prompted for a global update to all tables in the document comprising the database field. Again, as in the case of Figure 7A and 7B, a family table may be accessed in the database in order to determine the strategy in use for presentation of the graphical table(s) in the document. Since the graphical tables in this example may or may not belong to the same family the strategy selected for each table may or may not derive from the same family in the database. For example if one graphical table such as table 830 comprises Acme Laser printers and the subset of table 130 on the left of Figure 8B may comprise Laser printers, there may be different strategies for each graphical user interface table in terms of presentation. However, if the graphical table displayed to the user comprises no strategy and the parent table does comprise a strategy then the parent table strategy may be obtained from the family table and utilized in the child table. If the two tables in Figure 8B are siblings, then they may each comprise their own strategy as per the family table in the database or they one or both may utilize the parent family strategy for presentation depending on whether each particular sibling family has had a strategy for presentation set.

[0021] Figure 9A shows a subset of table 130 with a column to be resized by globally replacing text in at least one cell of the table. In this example all text "100-500-" is replaced with no text globally throughout the document or for example within the database. Figure 9B shows the resulting table columns sized to comprise the ratios of the columns in Figure 9A when for example the strategy employed or last set is "%" or ratio strategy. CONSTANT and VARIABLE table width mode settings per table or per document may be altered in order to control whether each table or all tables adjust in overall width or on a column basis.

[0022] Figure 10 shows a flow chart of the process of reallocating table formats in real-time. The process starts at 1000 with the user altering a table column via a

Graphical User Interface (GUI). This may involved inserting or deleting text or objects or dragging a column border to increase or decrease a column size. When the GUI delivers an event to the module utilized in embodiments of the invention to intercept a table column event at 1020, the module calculates the column widths for the remaining columns based on the strategy in use at 1030 as previous set or as queried to the user. The column widths of the columns are set to either keep the entire table width constant or to alter the sizes of the columns without regard for total table width using the strategy patterns described herein. If the global adjust for “N tables” at 1050 is set or as queried to the user, i.e., the scope of the adjustment is document scope, then the adjustment proceeds for “N tables” at 1060. If the global adjust option is not desired, then if the user is still adjusting a column (for example typing into a column or dragging a column) then the processing continues at 1010 or stops at 1080.

[0023] Thus embodiments of the invention directed to a method for reallocating table formats in real-time have been exemplified to one of ordinary skill in the art. The claims, however, and the full scope of any equivalents are what define the metes and bounds of the invention.

What is claimed is:

1. A method for reallocating table formats in real-time comprising:
altering a table column of a table via a user interface executing on a computing device
interacted with by a user;
intercepting a table column event generated with a module presenting said table via
said user interface wherein said table column event occurs as a result of said
altering said table column;
calculating a plurality of new column widths for all columns of said table based on all
column contents using a strategy pattern;
adjusting said plurality of new column widths on said user interface for said table;
and,
repeating said intercepting, said calculating and said adjusting while said user
continues said altering said table column.
2. The method of claim 1 wherein said strategy pattern comprises accessing a
parent family strategy pattern of a parent family.
3. The method of claim 1 wherein a first column selected from said table is set to
frozen mode and thereafter does not take part in said adjusting said plurality of
new column widths step.
4. The method of claim 1 wherein a first column selected from said table is set to
unfrozen mode and thereafter takes part in said adjusting said plurality of new
column widths step.
5. The method of claim 1 wherein said strategy pattern comprises determining an
amount of white space on each side of said all columns.
6. The method of claim 2 wherein said white space on each side of said all
columns is equal.
7. The method of claim 2 wherein said white space on each side of said all
columns is based on each item in each of said all columns.

8. The method of claim 1 wherein said strategy pattern comprises determining ratios between said all columns.
9. The method of claim 1 wherein said strategy pattern comprises calculating a function based on each item in each of said all columns and said ratios between said all columns.
10. The method of claim 9 wherein said strategy pattern comprises use of a minimum and maximum allowable column width parameter.
11. The method of claim 1 wherein said strategy pattern uses a table width mode during said calculating a plurality of new column widths wherein said table width mode is selected from the group consisting of VARIABLE and CONSTANT.
12. The method of claim 1 wherein said strategy pattern uses a scope during said calculating a plurality of new column widths wherein said scope is selected from the group consisting of table scope and document scope.
13. The method of claim 1 wherein said strategy pattern changes in real-time based on a user input.
14. The method of claim 13 wherein said user input comprises holding down a keyboard key.
15. The method of claim 13 wherein said user input comprises turning a scroll wheel.
16. The method of claim 1 wherein said altering comprises updating said table column from a database field.
17. The method of claim 1 further comprising altering all tables in a document comprising a particular element in at least one cell of said all tables.

18. The method of claim 1 wherein said altering comprises typing in said table column.
19. The method of claim 1 wherein said altering comprises deleting text from said table column.
20. The method of claim 1 wherein said altering comprises dragging said table column.
21. The method of claim 1 wherein said altering comprises pasting an object into said table column.
22. The method of claim 1 wherein said altering comprises deleting an object from said table column.
23. An apparatus for reallocating table formats in real-time comprising:
means for altering a table column of a table via a user interface executing on a computing device interacted with by a user;
means for intercepting a table column event generated with a module presenting said table via said user interface wherein said table column event occurs as a result of said means for altering said table column;
means for calculating a plurality of new column widths for all columns of said table based on all column contents using a strategy pattern;
means for adjusting said plurality of new column widths on said user interface for said table; and,
means for repeating said means for intercepting, said means for calculating and said means for adjusting while said user operates said means for altering said table column.
24. The apparatus of claim 23 wherein said strategy pattern comprises means for accessing a parent family strategy pattern of a parent family.
25. The apparatus of claim 23 wherein said means for altering comprises means for

updating said table column from a database field.

26. The method of claim 25 further comprising means for altering all tables in a document comprising a particular element in at least one cell of said all tables.

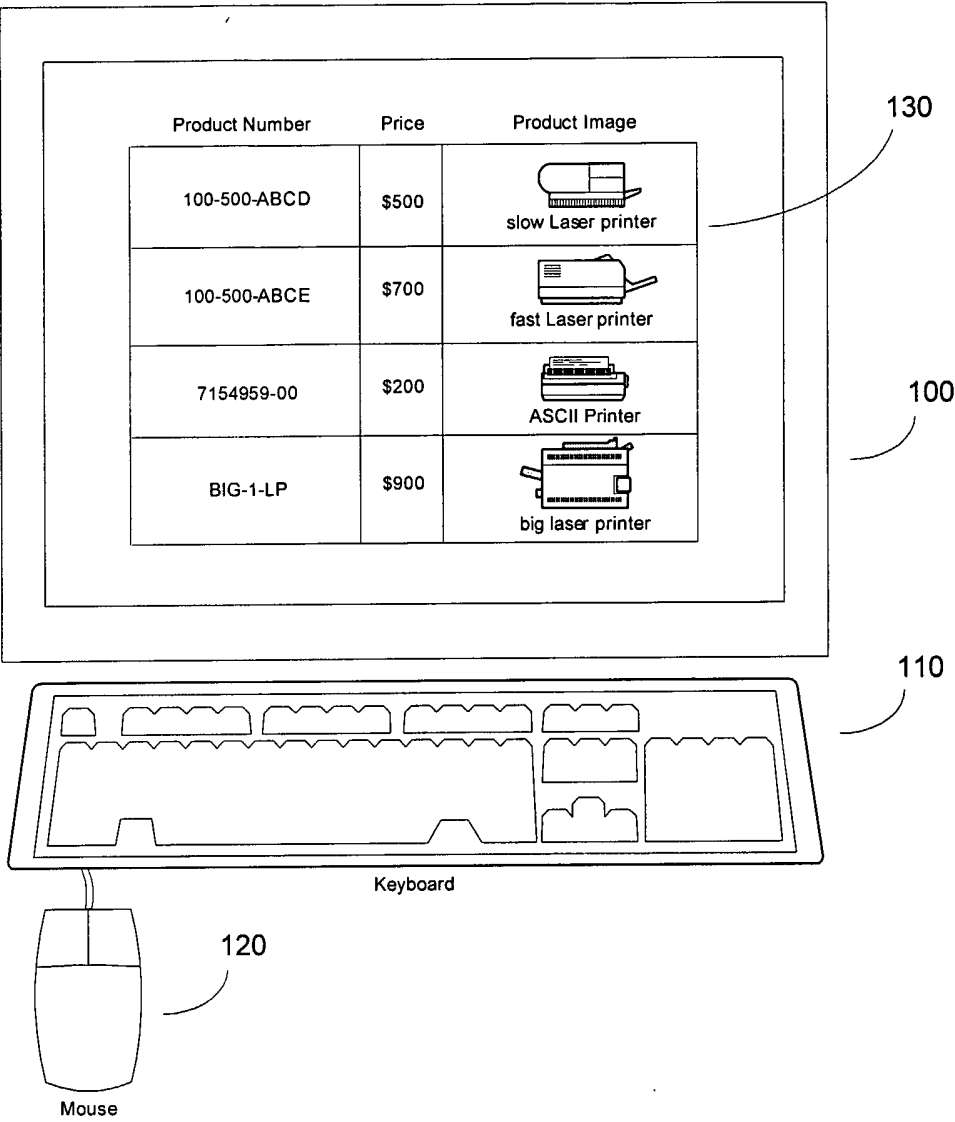


FIGURE 1

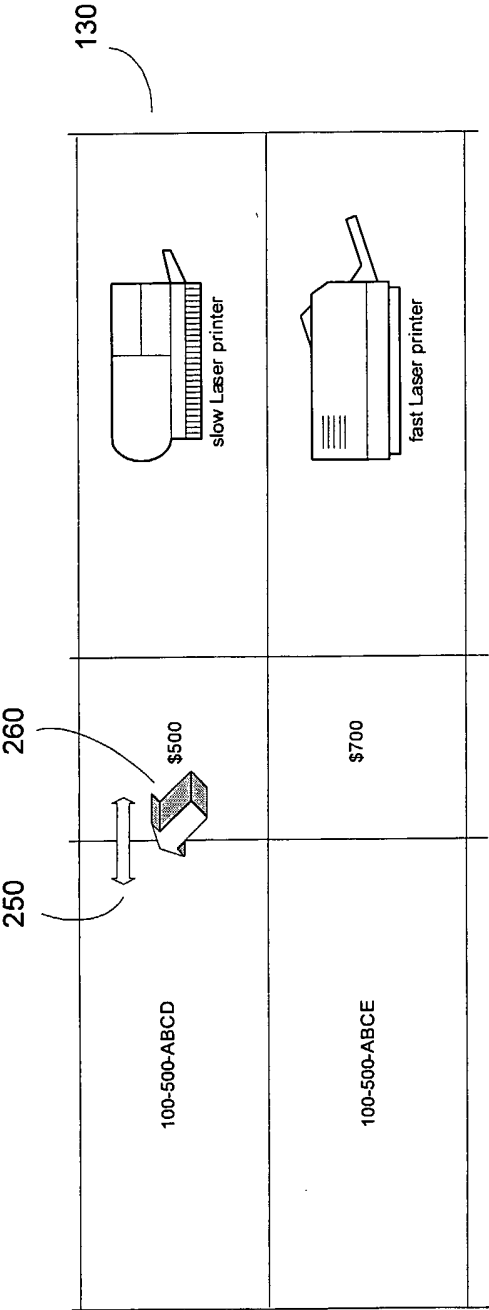


FIGURE 2A

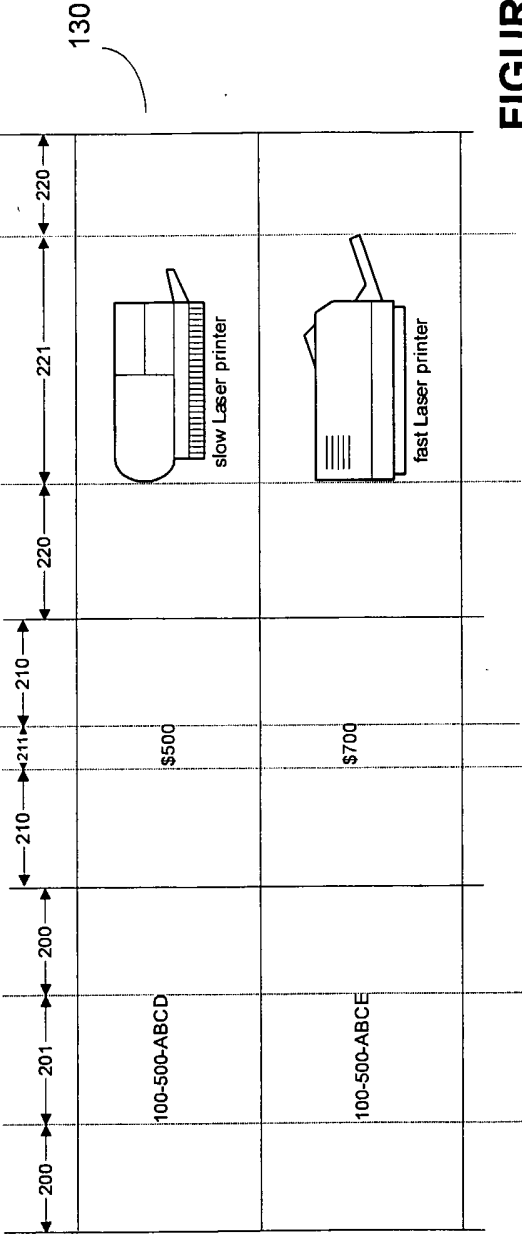


FIGURE 2B

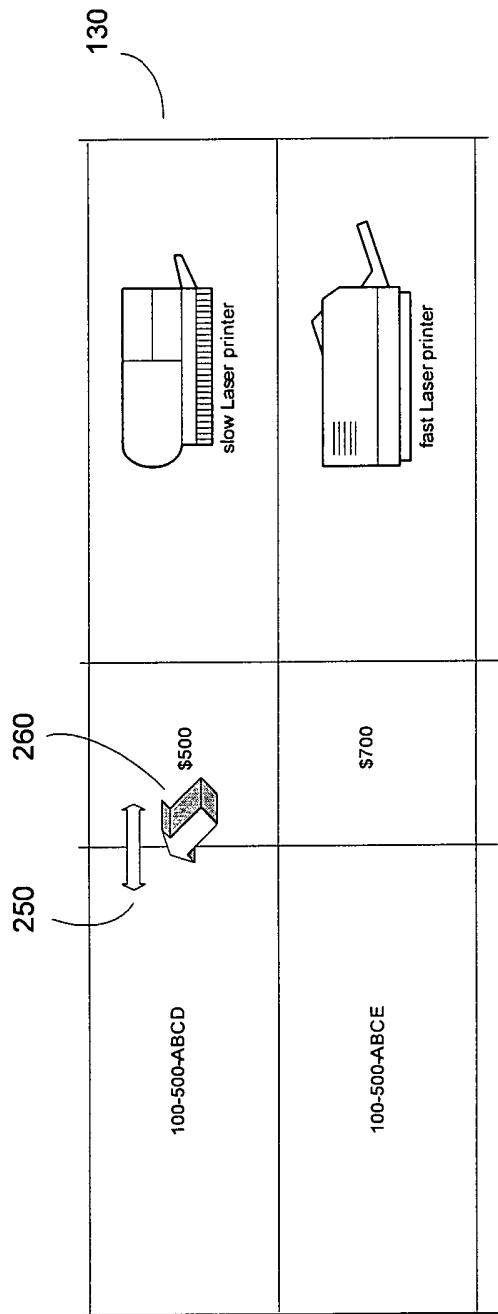


FIGURE 2C

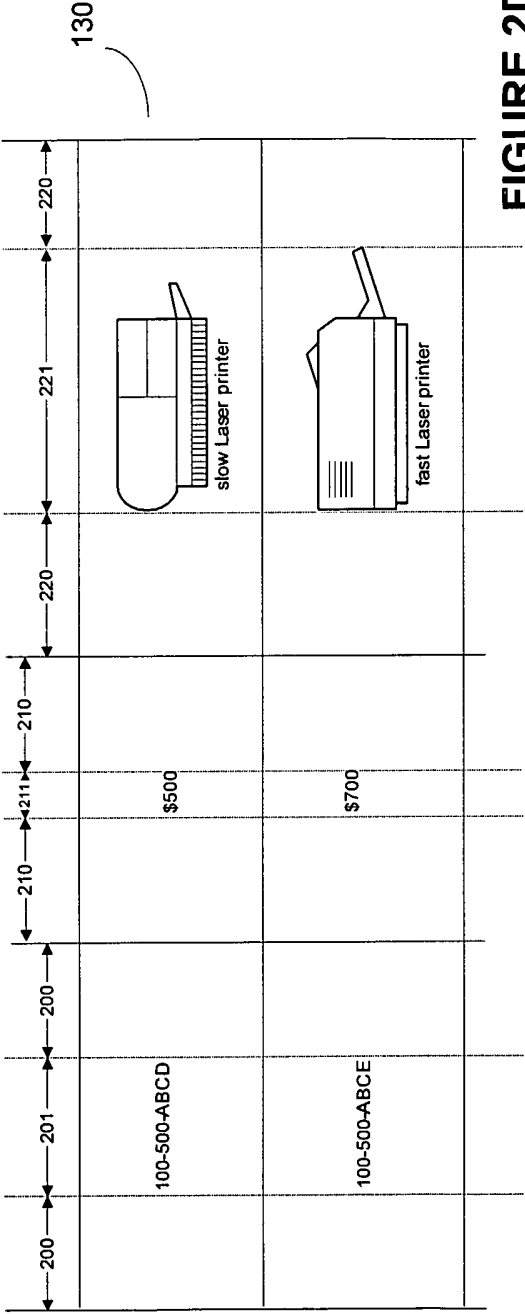


FIGURE 2D

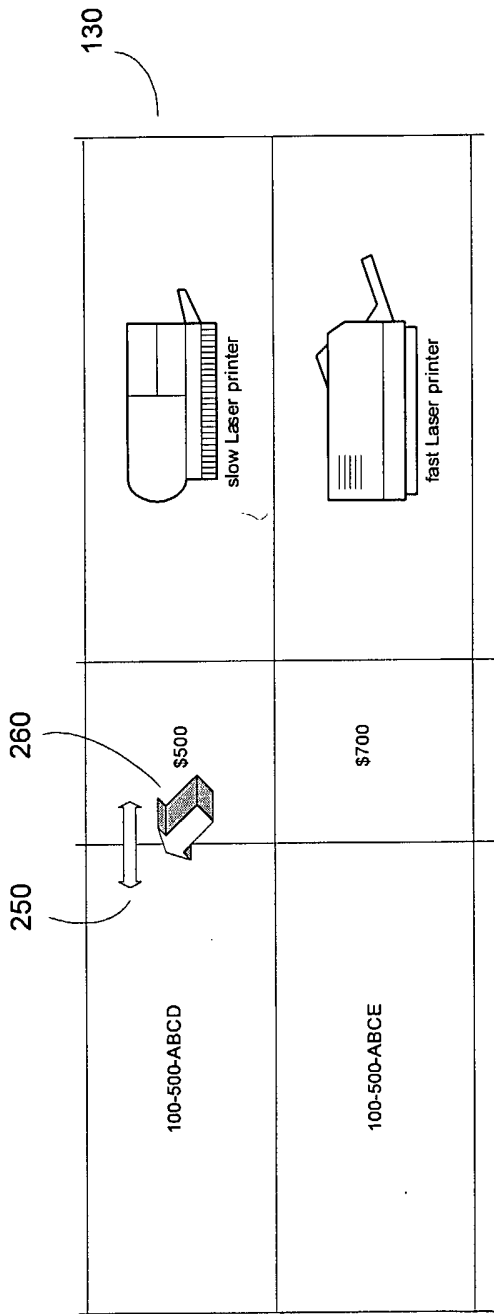


FIGURE 3A

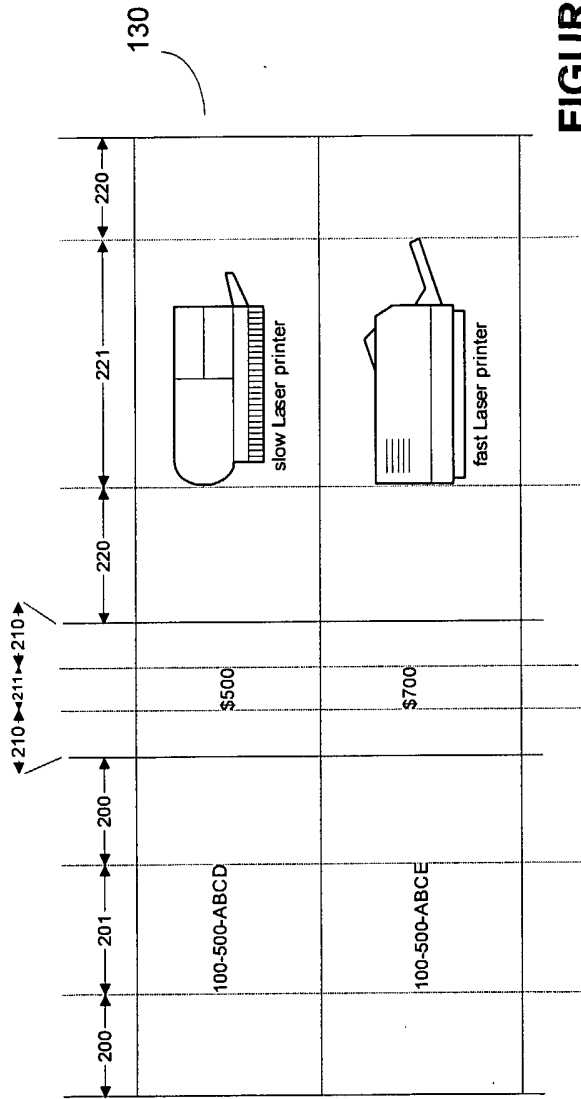


FIGURE 3B

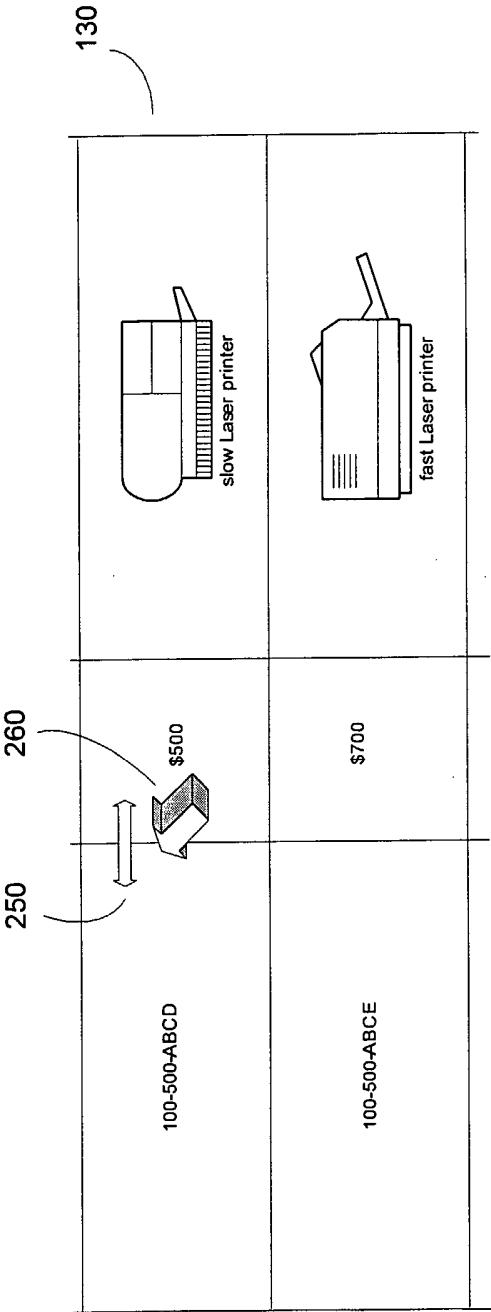


FIGURE 4A

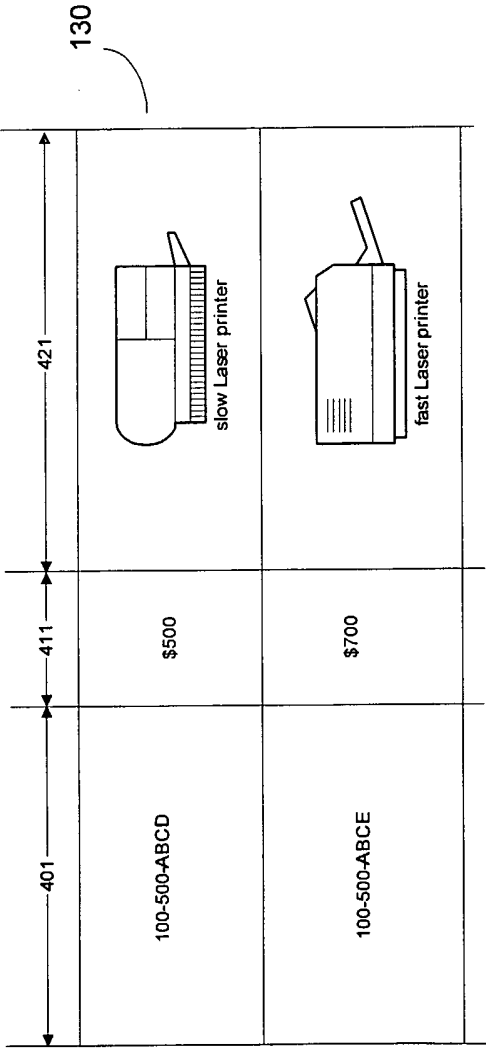


FIGURE 4B

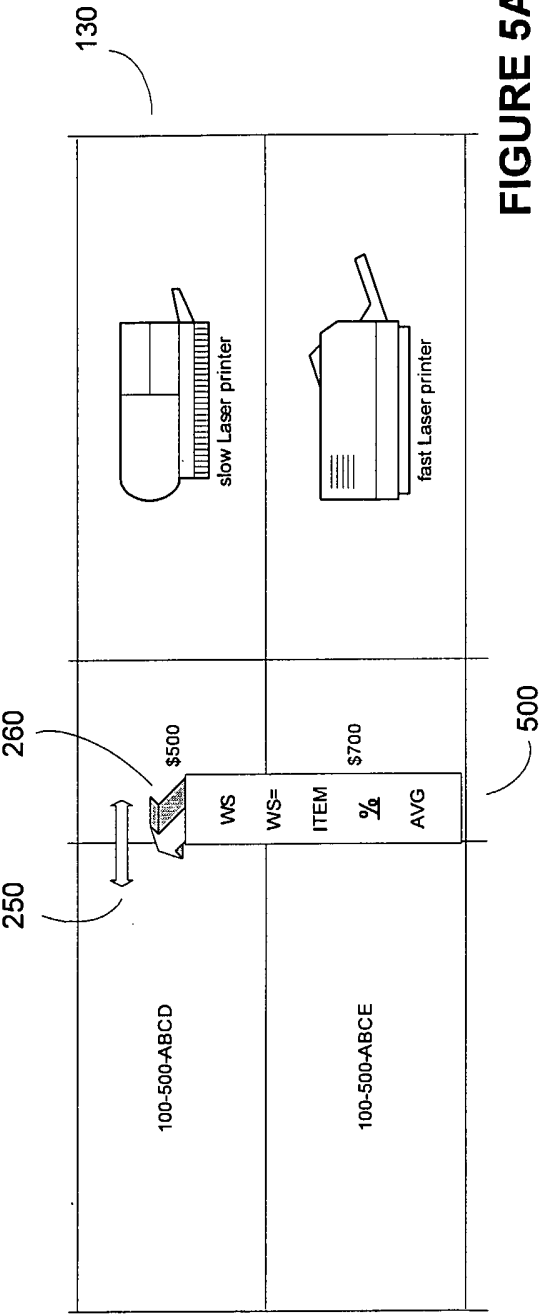


FIGURE 5A

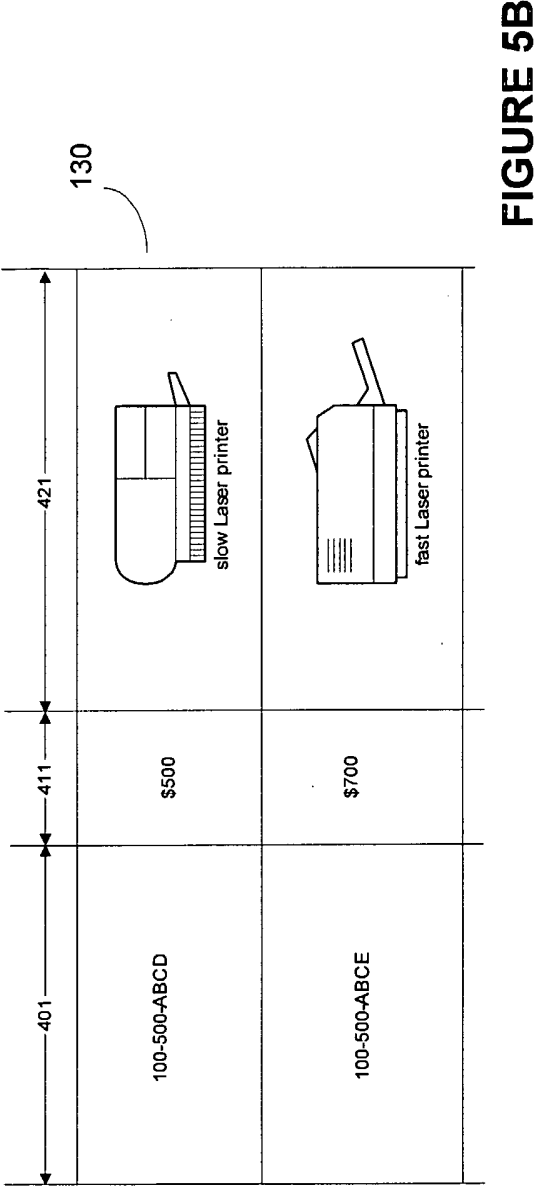
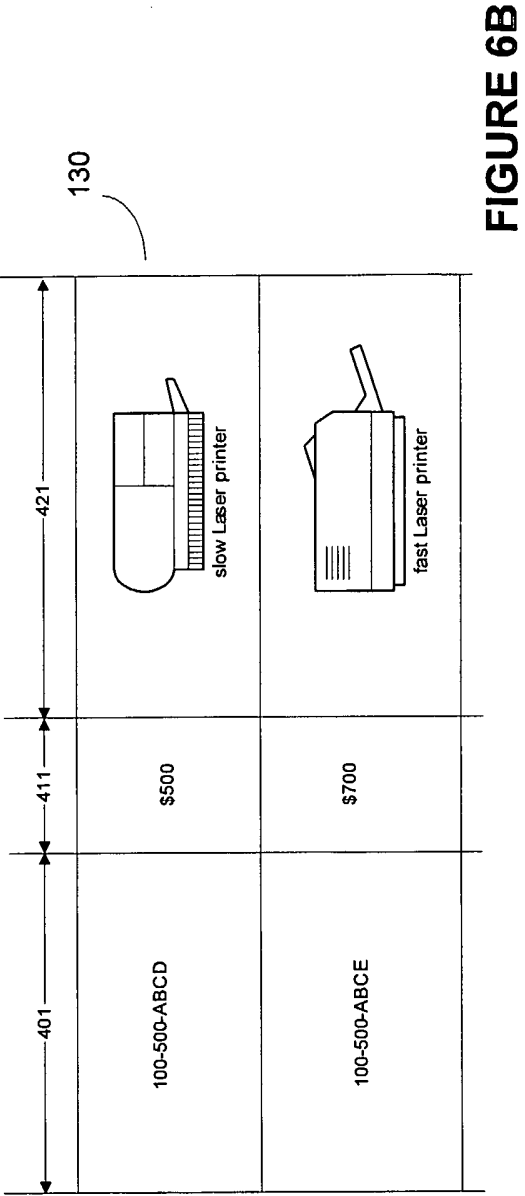
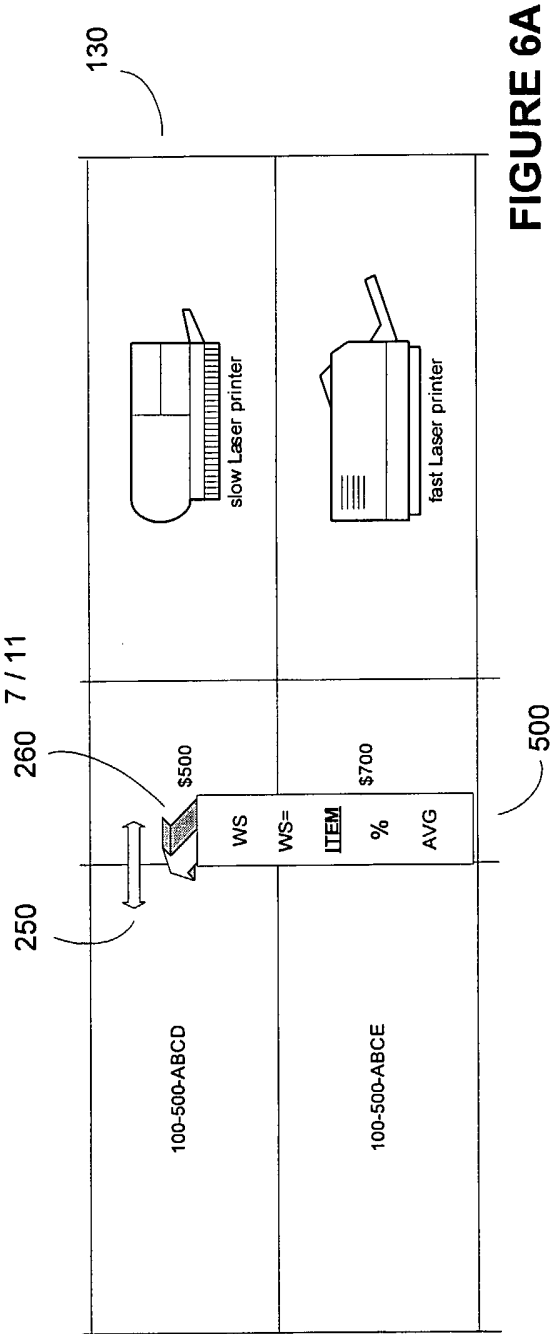


FIGURE 5B



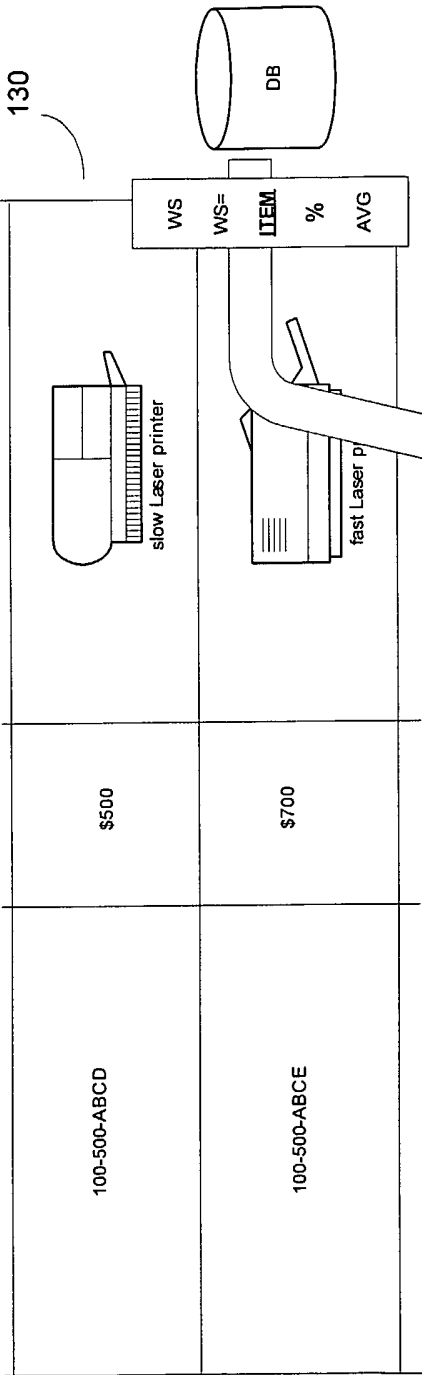


FIGURE 7A

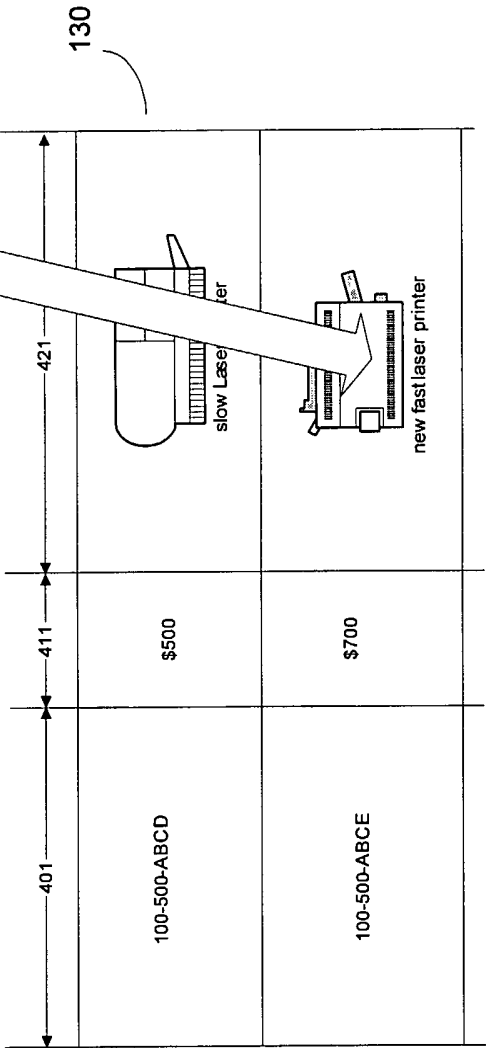


FIGURE 7B

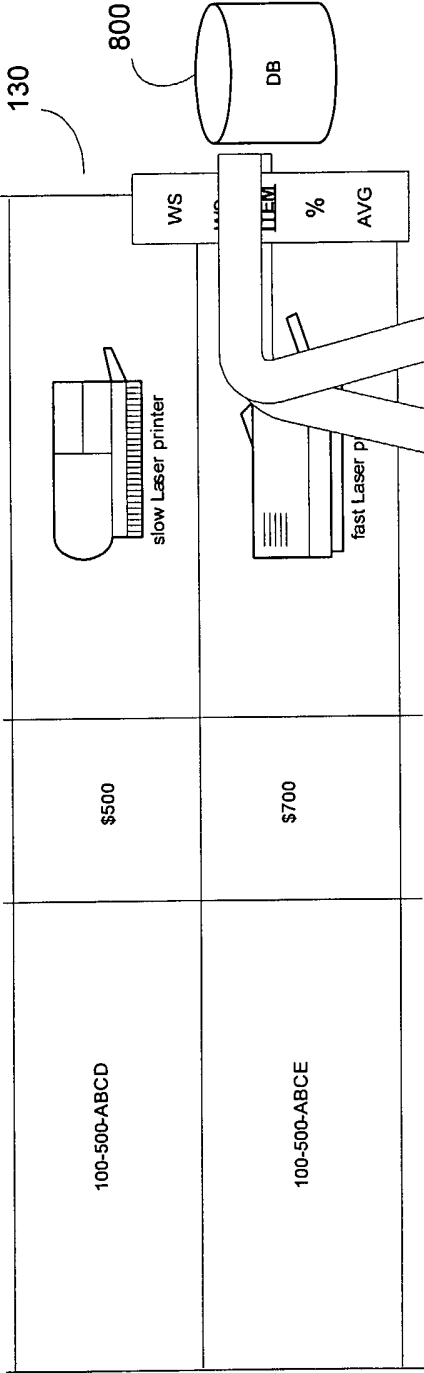


FIGURE 8A

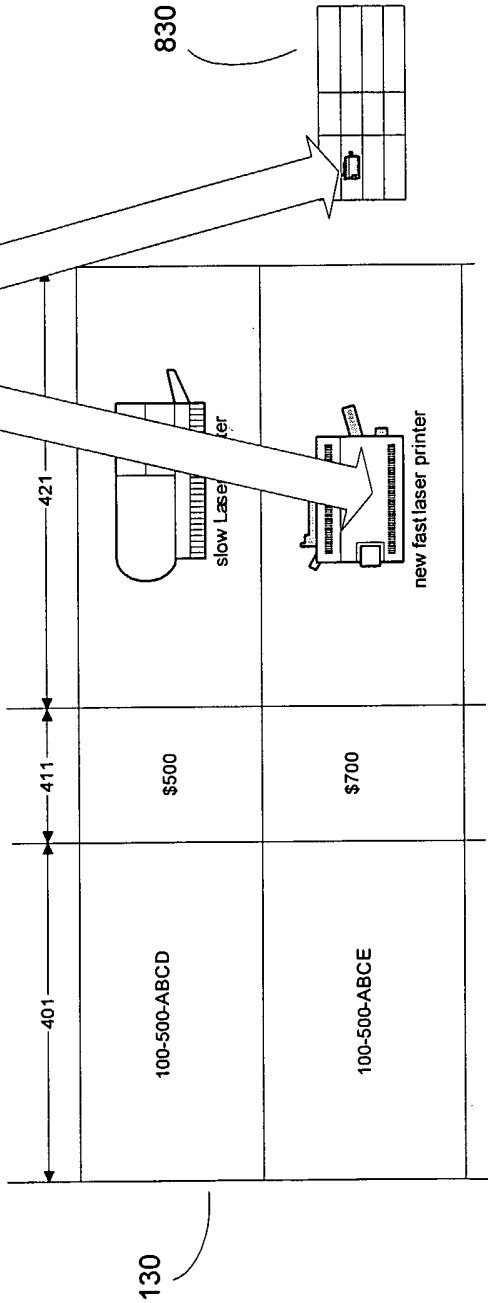


FIGURE 8B

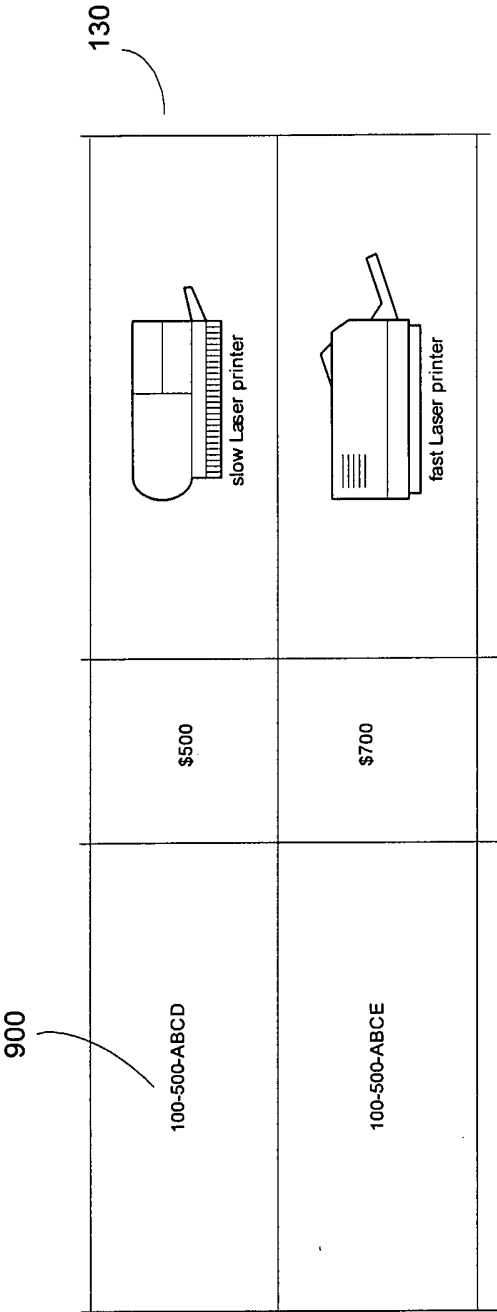


FIGURE 9A

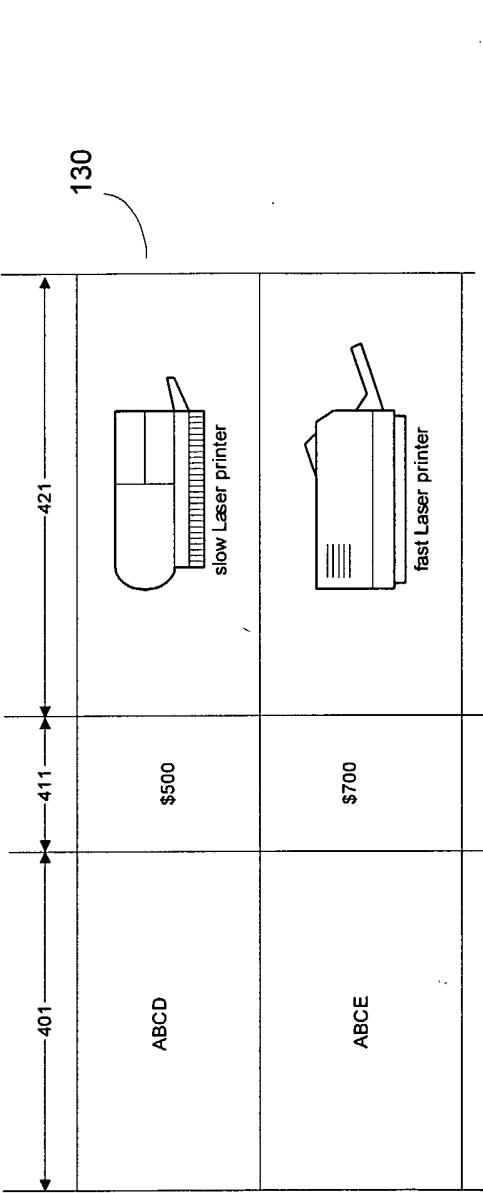
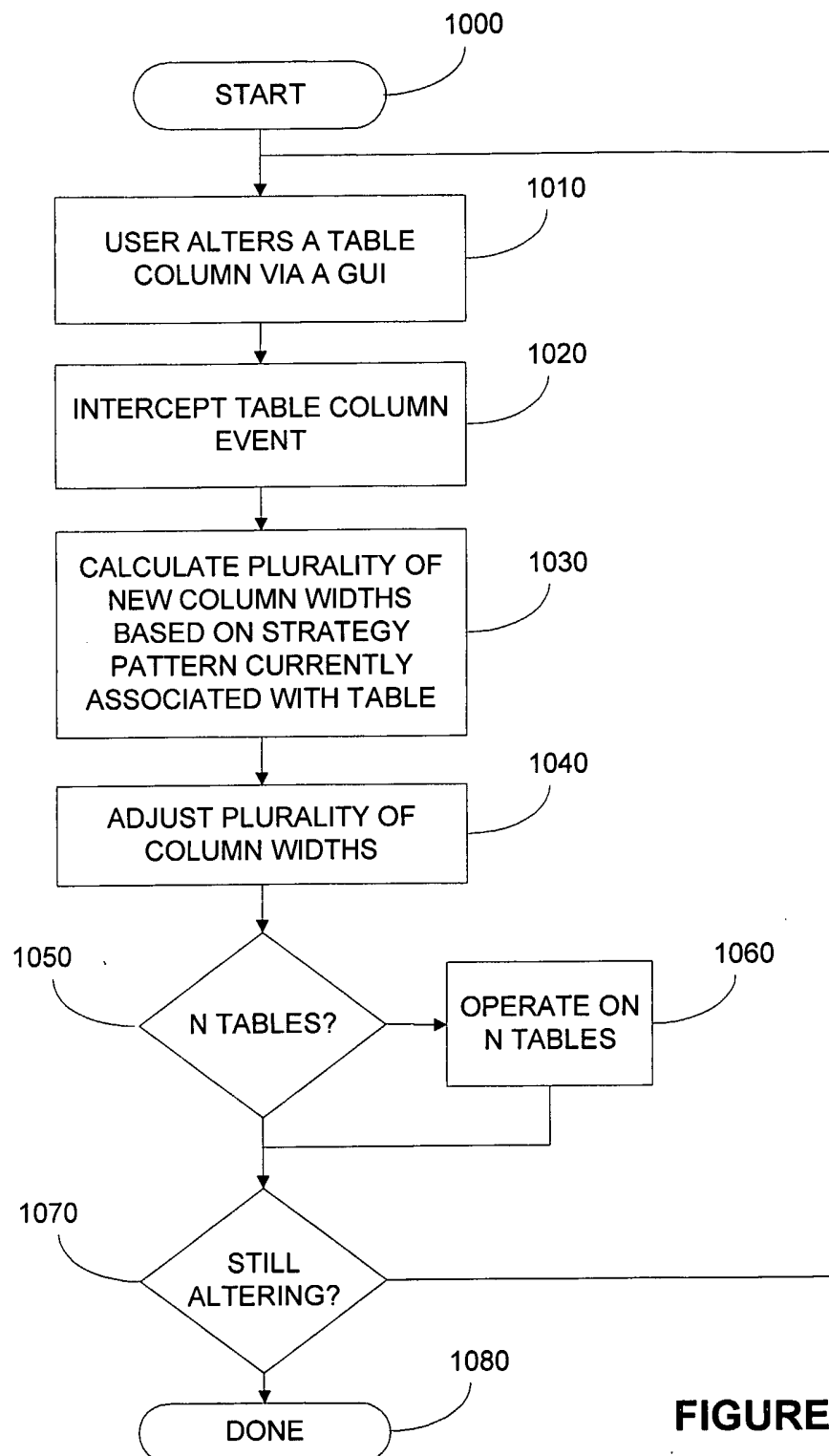


FIGURE 9B

**FIGURE 10**