



US 20130325638A1

(19) **United States**(12) **Patent Application Publication**
Auclair et al.(10) **Pub. No.: US 2013/0325638 A1**(43) **Pub. Date: Dec. 5, 2013**(54) **METHOD, SYSTEM AND COMPUTER
PROGRAM FOR ASSIGNING AN
ASSORTMENT OF PRODUCTS TO AN
EXISTING PLANOGRAM**(52) **U.S. Cl.**CPC **G06Q 30/0276** (2013.01)USPC **705/14.72**(71) Applicant: **CVDM-Solutions**, Paris (FR)(72) Inventors: **Adrien Auclair**, Paris (FR); **Adrien
Guillerot**, Paris (FR)(21) Appl. No.: **13/909,259**(22) Filed: **Jun. 4, 2013****Related U.S. Application Data**(60) Provisional application No. 61/654,959, filed on Jun.
4, 2012.**Publication Classification**(51) **Int. Cl.**
G06Q 30/02 (2006.01)(57) **ABSTRACT**

Method for assigning an assortment of products to an existing planogram includes: providing an existing planogram which includes an existing assortment of products in association with a geometrical placement layout of these products on shelves and storing the existing planogram in memory; providing a user-defined new assortment of products and storing the new assortment in memory; furnishing, by a processor having read/write access to the memory, an output planogram by removing, from the output planogram, products in the existing assortment but not in the new assortment and adding, to the output planogram, products in the new assortment but not in the existing assortment; storing the output planogram in memory. The method further includes: generating a set of at least one segmentation tree which describes the relative geometrical placement of products in the existing planogram according to several ordered category types, each type corresponding to a level of the segmentation tree.

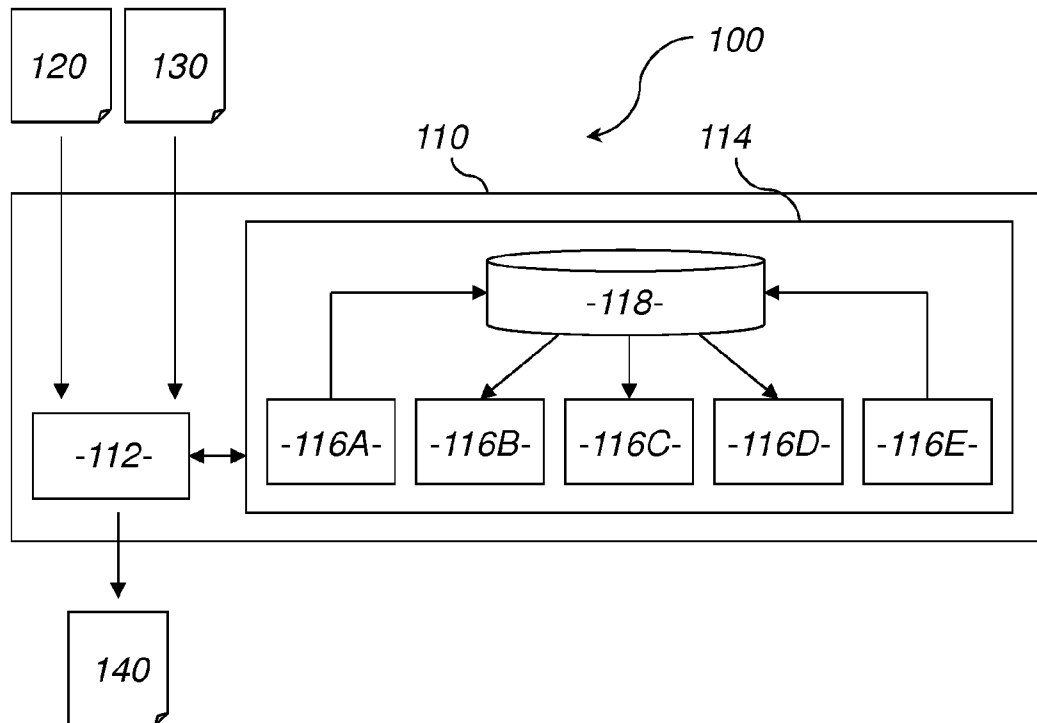


Figure 1

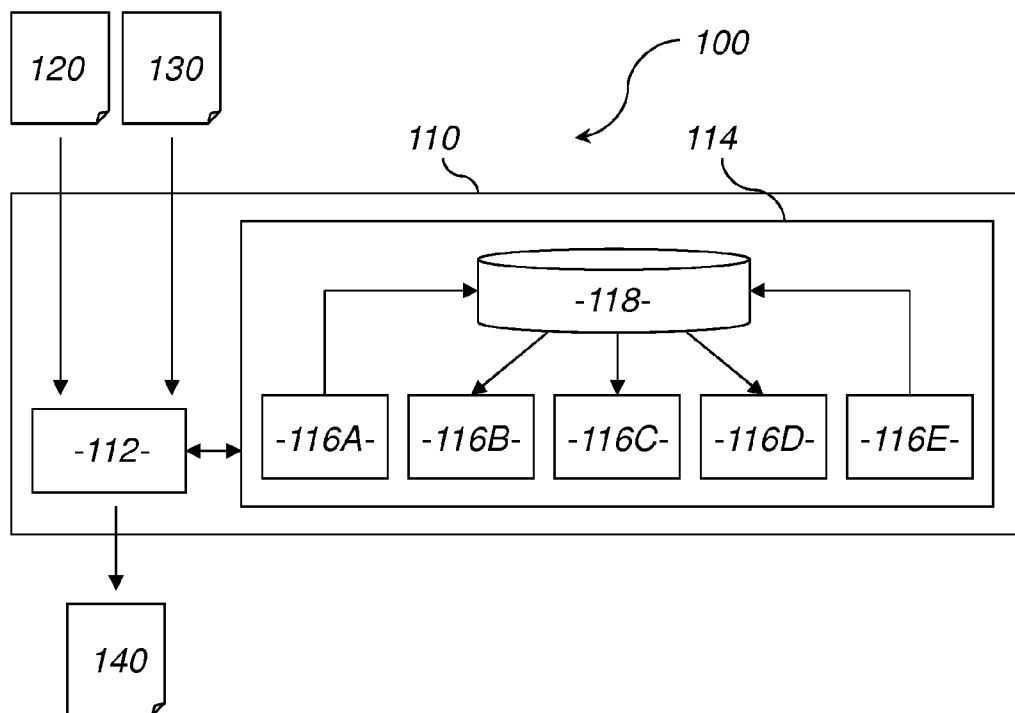
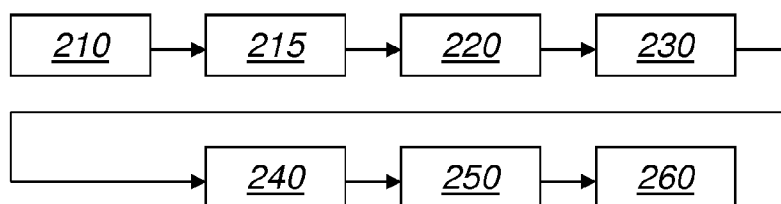


Figure 2



120 Figure 3



P	C ₁ (P) « Segment »	C ₂ (P) « Brand »	n _a (P)
145	S1	A	4
223	S1	B	1
349	S1	C	2
7821	S2	A	1
7412	S2	A	1
8745	S2	B	1
8462	S2	B	2
0122	S3	A	2
0199	S3	A	2
0481	S3	A	1
1411	S3	B	3
2010	S3	C	1
2100	S3	C	1
2610	S3	C	2
9111	S3	D	4
9784	S4	D	1
9785	S4	D	3
9786	S4	D	1

Figure 4

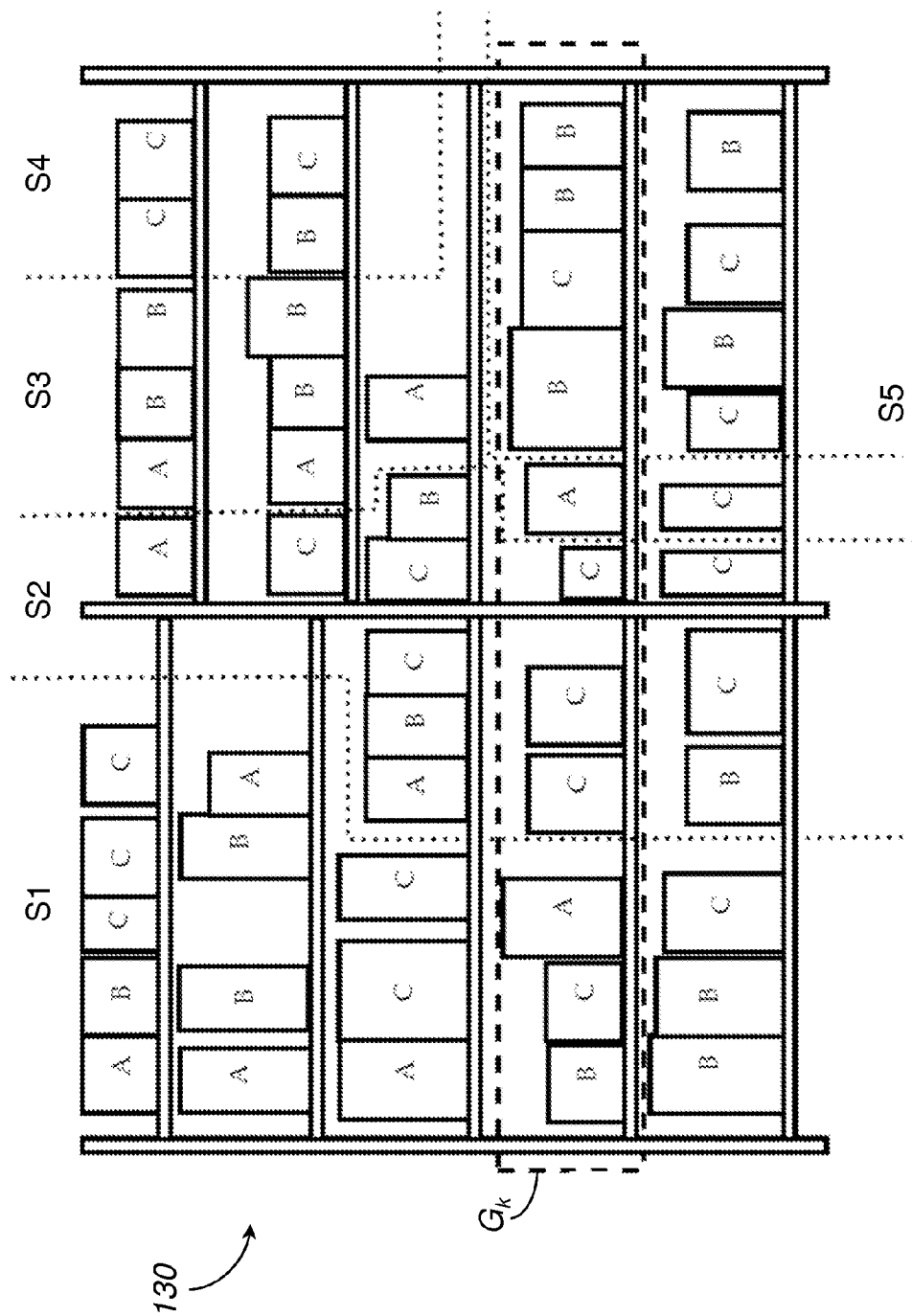


Figure 5

220

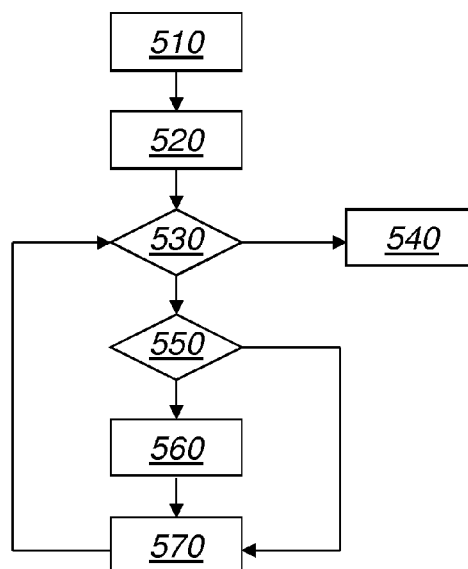


Figure 6

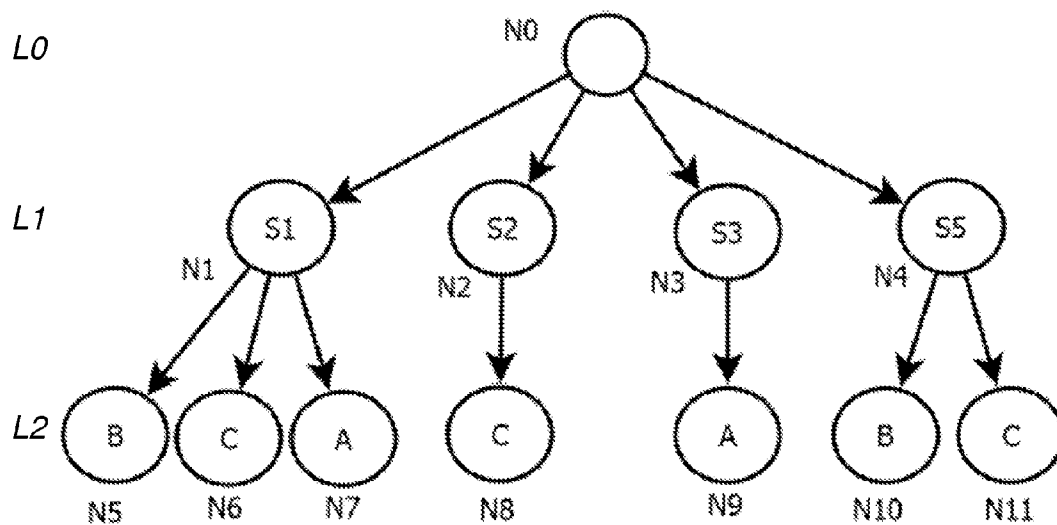


Figure 7

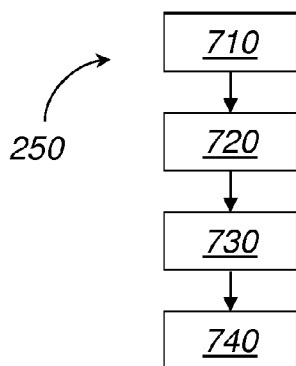


Figure 8

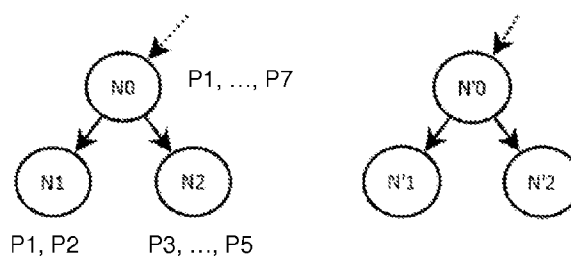
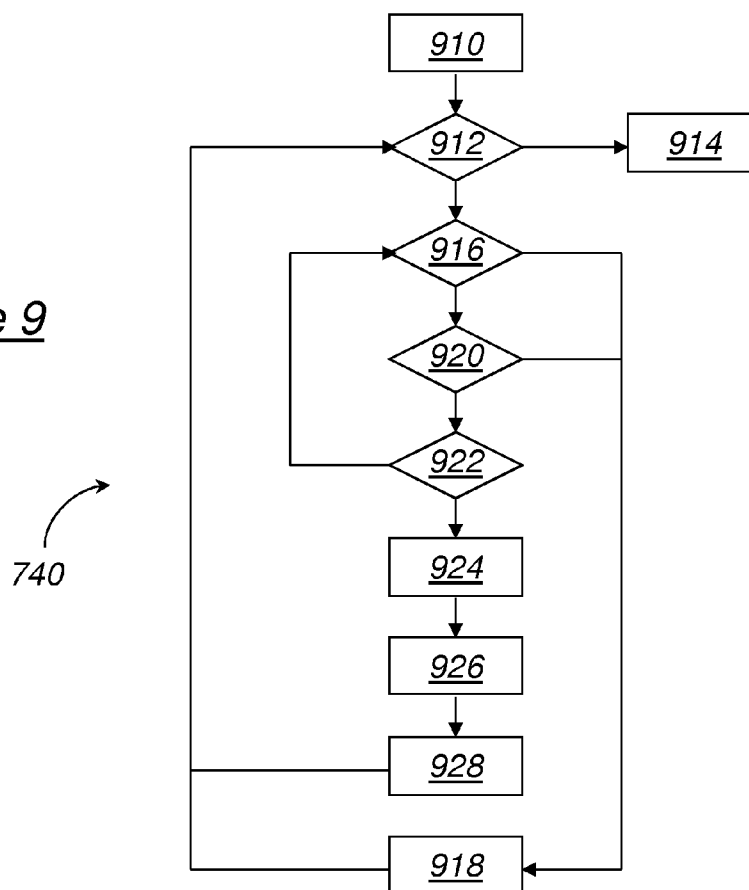


Figure 9



METHOD, SYSTEM AND COMPUTER PROGRAM FOR ASSIGNING AN ASSORTMENT OF PRODUCTS TO AN EXISTING PLANOGRAM

BACKGROUND OF THE INVENTION

[0001] It relates generally to the field of retail merchandising, and more specifically to the automated production of store-specific product placement within shelving areas using an existing planogram and a corresponding existing assortment of products.

[0002] A planogram can be defined as a diagram or model that indicates the geometrical placement of retail products on shelves in order to maximize sales. Therefore, a planogram includes an assortment of products in association with a geometrical placement layout of these products on shelves.

[0003] Conventional space and category management and planning systems are used to specify the location and placement of products within a retail space or store, such as, for example, a supermarket, a consumer electronic retail store, or the like. Retailers expend great time and effort in considering where and how to place products on the limited store shelves. The profitability of the store is in part dependent on an optimal placement of products for perusal by customers. If a customer cannot find a product, or a product does not catch his or her eyes, or if there is insufficient stock on a shelf to meet demand, then a sale may be lost. Retailers must make products available, appealing, and easy to find in order to maximize sales. Merchandising, in the sense of product placement, must be continuously updated with shifts in consumer buying habits, seasonal rotation, and new product offerings.

[0004] Manufacturers and distributors compete intensely for desirable shelving locations and maximum facings (number of rows of product facing the customer). Retailers are careful to place high-volume, high-profit-margin products in preferred locations, i.e., front of store, end of aisles, and eye-level shelves. With finite shelving area, each product must be allocated some number of facings to satisfy consumer demand, minimize shelf-stocking labor, and maximize visible exposure. The process of maximizing sales and profitability, given the constraints of limited shelving area and competition among suppliers for the best shelves and facings, is a very difficult, yet important issue for retailers.

[0005] A major space-management problem facing retailers and suppliers today is poor implementation of merchandising plans and an inability to react quickly to changes in consumer demand with revised shelf layouts. As mentioned above, a planogram is a product-placement layout, generated by a planning tool or computer program, which defines where products will be placed on the available store shelves, both in terms of shelf location and number of facings. Large retail chains routinely generate global or cluster planograms for local stores to implement. However, such generic planograms, which are designed to be shared across several stores, often do not accurately reflect the specific in-store fixture equipment or local customer demand for products. The global planogram may need to be adapted to the specific needs of the local store. The variability of shelving areas and/or of the product assortment between different stores often requires manual modification of the layout by the person stacking the shelf, in order to make everything fit.

[0006] Historically, planograms have been done manually, often using an existing merchandise planogram as a template.

The planogram designer starts with the existing planogram and then makes the necessary changes to accommodate the new shelving format and/or product placement and/or product assortment. Other automated techniques divide the products into logical groups and then put products in sequence from top to bottom, snaking left to right across each shelf within defined horizontal limits. However, if the target product assortments or configurations vary too much between the planogram to be updated and the template, the template effectively becomes unusable. A new planogram has to be created, which defeats the purpose of the template; i.e., in reality the planogram must be started from the beginning.

[0007] Retailers continuously look for a competitive edge. A significant opportunity lays in the ability to customize the configuration of products to be more targeted to the particular store, its local specifications, product assortment evolution, and fluctuations in consumer demand. A need exists to readily adapt planograms for product placement that are customized to particular needs.

[0008] Therefore, the invention more particularly applies to a method for assigning an assortment of products to an existing planogram which comprises the following steps:

[0009] providing with an existing planogram which includes an existing assortment of products in association with a geometrical placement layout of these products on shelves and storing said existing planogram in memory,

[0010] providing with a user-defined new assortment of products and storing said new assortment in memory,

[0011] furnishing, by a processor which has read/write access to the memory, an output planogram by removing, from the output planogram, products which are in the existing assortment but not in the new assortment and by adding, to the output planogram, products which are in the new assortment but not in the existing assortment, and

[0012] storing said output planogram in memory.

DESCRIPTION OF THE PRIOR ART

[0013] Several solutions have been proposed to automatically generate planograms specifically adapted to retail stores. For example, in US patent application publication No. US 2003/0154141 A1, a global inventory management system is presented, in which the planogram representing the current placement of the products may be automatically optimized for stock management. However, no specific method to achieve this goal is described.

[0014] In US patent application publication No. US 2006/0149634 A1 and in US patent application publication No. US 2012/0022913 A1, systems are presented to generate planograms or assortments of products, that are optimized for product placement, and using a variety of product data. However, they do not precisely describe a specific method for planogram generation.

[0015] In US patent application publication No. US 2005/0203790 A1, a planogram is automatically generated for a given retail store using input specific store data and a set of business rules. However, business rules are often subject to change and may even depend on the store. So the use of business rules may not always be practical.

[0016] In US patent application publication No. US 2007/0288296 A1, a planogram is automatically generated using an initial arrangement of products in a two-dimensional grid structure, and products are placed on a planogram compatible

with store data after creating a prioritized list of products. It is a common feature of the last two cited patent application publications that they do not take advantage of a previous placement of products when generating a planogram specific to a store.

[0017] It may thus be desirable to provide a method for assigning an assortment of products to an existing planogram that at least partly dispenses with the aforementioned problems and constraints and improves on the existing methods.

SUMMARY OF THE INVENTION

[0018] The present invention intends to provide a method to apply an assortment of products to a user-input existing planogram that specifically addresses the need of merchandising industries to apply a common assortment of products to existing planograms, so that planograms are customized to the needs of each particular retail store. In one embodiment of the present invention, a method for assigning an assortment of products to an existing planogram includes receiving user input specifying a new assortment of products and an existing planogram containing any set of products (i.e. an existing assortment of products) intersecting or not intersecting the set of products in the input new assortment. The method according to the invention then includes a step where a set of at least one segmentation tree which describes the relative geometrical placement of products in the existing planogram is created. The present invention may then allow the deletion of products in the existing planogram that are not in the new assortment. The method includes using the generated set of at least one segmentation tree mentioned previously to add products in the new assortment but not in the existing planogram.

[0019] It is therefore proposed a method for assigning an assortment of products to an existing planogram which comprises the following steps:

[0020] providing with an existing planogram which includes an existing assortment of products in association with a geometrical placement layout of these products on shelves and storing said existing planogram in memory,

[0021] providing with a user-defined new assortment of products and storing said new assortment in memory,

[0022] furnishing, by a processor which has read/write access to the memory, an output planogram by removing, from the output planogram, products which are in the existing assortment but not in the new assortment and by adding, to the output planogram, products which are in the new assortment but not in the existing assortment,

[0023] storing said output planogram in memory, wherein the method further comprises:

[0024] generating, by the processor, a set of at least one segmentation tree which describes the relative geometrical placement of products in the existing planogram according to several ordered category types, each category type corresponding to a level of said at least one segmentation tree,

[0025] adding the new products to the output planogram using the generated set of at least one segmentation tree.

[0026] Accordingly advantages of one or more aspects of the invention are as follows: providing a method for automatically adapting planograms without the need to enumerate difficult business rules and without the need to build template planograms or arrangements of products in a grid structure

for the implementation of possibly implicit business rules, and for generating output planograms from an input assortment of products that are well adapted to store specific data. Generating and using a tree structure of the existing planogram makes it possible to automatically add new products at optimal places in the output planogram, and with optimal number of facings.

[0027] Optionally, adding new products to the output planogram using the generated set of at least one segmentation tree further uses optimization criteria for placement and number of facings of products on the shelves.

[0028] Optionally also, the ordered category types include at least a "segment" type which defines several geometrically segmented areas of the shelves and a "brand" type which defines several brands to which every product can be associated.

[0029] Optionally also, the output planogram is the existing planogram from which products which are in the existing assortment but not in the new assortment are directly removed and to which products which are in the new assortment but not in the existing assortment are directly added.

[0030] Optionally also, generating said at least one segmentation tree includes:

[0031] partitioning the existing planogram into several groups, wherein each group contains one or more shelves located at the same height in the existing planogram,

[0032] generating one segmentation tree for each group.

[0033] Optionally also, generating a segmentation tree includes creating a root node that identifies this segmentation tree and iterating over the corresponding products of the existing planogram as follows:

[0034] for each product, determine the corresponding category values taken for each category type, and

[0035] if there exists no node in the segmentation tree at any level such that this node and its ancestors are associated to at least part of the category values determined for the product, then create such a new node with all necessary ancestors from the root node to complete a corresponding branch of the segmentation tree and update geometrical information associated to said new node and its ancestors,

[0036] else, update geometrical information associated to said associated node and its ancestors.

[0037] Optionally also, geometrical information is updated using placement data of the product, such as the identification of the shelf on which the product is placed, the position of the product on this shelf and the length of linear the product occupies.

[0038] Optionally also, adding the new products to the output planogram using the generated set of at least one segmentation tree includes:

[0039] creating a subset of products in the new assortment but not in the existing assortment of existing planogram,

[0040] associating each product of the subset to all nodes of any segmentation tree with compatible category values, and

[0041] iteratively adding each product of the subset to the output planogram in association with an available node and updating geometrical information associated with said available node.

[0042] It is further proposed a system for assigning an assortment of products to an existing planogram comprising:

[0043] a memory for storing:

[0044] an existing planogram which includes an existing assortment of products in association with a geometrical placement layout of these products on shelves,

[0045] a user-defined new assortment of products, and

[0046] an output planogram;

[0047] a processor which has read/write access to the memory for furnishing the output planogram by:

[0048] removing, from the output planogram, products which are in the existing assortment but not in the new assortment,

[0049] adding, to the output planogram, products which are in the new assortment but not in the existing assortment,

wherein the processor has read/write access to the memory for further:

[0050] generating a set of at least one segmentation tree which describes the relative geometrical placement of products in the existing planogram according to several ordered category types, each category type corresponding to a level of said at least one segmentation tree,

[0051] adding the new products to the output planogram using the generated set of at least one segmentation tree.

[0052] It is further proposed a computer program that can be downloaded from a communication network and/or saved on a computer-readable medium and/or executed by a processor, said program comprising program code instructions for the execution of the steps of a method for assigning an assortment of products to an existing planogram as defined previously, when said program is executed on a computer.

BRIEF DESCRIPTION OF THE DRAWINGS

[0053] The invention will be better understood by means of the following description, given solely by way of example and made with reference to the accompanying drawings, in which:

[0054] FIG. 1 illustrates a computer system with a software module for assigning an assortment of products to an existing planogram,

[0055] FIG. 2 illustrates the successive steps of a method for assigning an assortment of products to an existing planogram, according to a preferred embodiment of the invention,

[0056] FIG. 3 illustrates an example new assortment of products as input data for the execution of the method in FIG. 2,

[0057] FIG. 4 illustrates an example existing planogram as input data for the execution of the method in FIG. 2,

[0058] FIG. 5 illustrates details of a step of the method in FIG. 2, for generating a set of at least one segmentation tree,

[0059] FIG. 6 illustrates an example segmentation tree constructed from the existing planogram illustrated in FIG. 4, after execution of the step in FIG. 5

[0060] FIG. 7 illustrates an example method for adding new products to an output planogram using a set of at least one segmentation tree,

[0061] FIG. 8 illustrates two example subtrees with similar nodes,

[0062] FIG. 9 illustrates details of a step of the method in FIG. 7, for adding products of a new assortment of products to an existing planogram.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0063] FIG. 1 illustrates a possible embodiment of a system 100 according to the invention for assigning an assortment of products to an existing planogram. This system 100 includes an assignment module 110 that receives input from a user. The input given by the user contains at least an assortment 120 of products and a planogram 130. The planogram 130 is an existing planogram which includes an existing assortment of products in association with a geometrical placement layout of these products on shelves. In a possible embodiment, the existing planogram 130 is specific to a targeted retail store and may represent the actual observed product placement in the store. The assortment 120 is a new assortment of products intersecting or not intersecting the existing assortment of products in the existing planogram 130.

[0064] The assignment module 110 processes the inputs 120 and 130 to generate an output planogram 140 which is a product-placement layout where products are taken from the new assortment 120 and where planogram geometrical constraints specific to the store are taken from the existing planogram 130.

[0065] The assignment module 110 schematically represented in FIG. 1 has a central processing unit 112 associated with a memory 114 (for example a RAM memory).

[0066] The assignment module 110 can be implemented as a software module in a computer device, such as a conventional computer that includes a microprocessor, i.e. the central processing unit 112, associated with one or more memories, i.e. the memory 114, for storing data files and instructions that are executed by the microprocessor as computer programs.

[0067] Therefore, the assignment module 110 as illustrated in FIG. 1 functionally stores five computer programs 116A, 116B, 116C, 116D and 116E or five functions of the same computer program in the memory 114. Also note that the computer programs 116A, 116B, 116C, 116D and 116E are shown as being separate, but this separation is purely functional. They may just as well be grouped into one or more software programs according to any combination. Their functions could also be at least partly micro-programmed or micro-wired in dedicated integrated circuits. Thus, alternatively, the computer device implementing the assignment module 110 could be replaced by an electronic device comprised solely of digital circuits (without a computer program) for carrying out the same actions.

[0068] The memory 114 further comprises a dedicated zone 118 for storing input and output data for and from the execution of the computer programs 116A, 116B, 116C, 116D and 116E, such as the new assortment 120, the existing planogram 130 and the output planogram 140.

[0069] The first computer program 116A includes program code instructions for the execution of storing the existing planogram 130 and the new assortment 120 in the dedicated zone 118 of the memory 114.

[0070] The second computer program 116B includes program code instructions for the execution of generating a set of at least one segmentation tree which describes the relative geometrical placement of products in the existing planogram 130 according to several ordered category types, each category type corresponding to a level of said at least one segmentation tree. These category types correspond to categories of the products both in the existing and new assortments and in the existing and output planograms. They may include at

least a “segment” type which defines several geometrically segmented areas of the shelves and a “brand” type which defines several brands to which every object can be associated. In a possible embodiment, the assignment module 110 may additionally receive the ordering information of these category types as input from the user.

[0071] The third computer program 116C includes program code instructions for the execution of removing, from the output planogram 140, products which are in the existing assortment of the existing planogram 130 but not in the new assortment 120.

[0072] The fourth computer program 116D includes program code instructions for the execution of adding, to the output planogram 140, products which are in the new assortment 120 but not in the existing assortment of the existing planogram 130, wherein adding the new products to the output planogram 140 uses the generated set of at least one segmentation tree.

[0073] Finally, the fifth computer program 116E includes program code instructions for the execution of storing the output planogram 140 in the dedicated zone 118 of the memory 114.

[0074] The method for assigning an assortment of products to an existing planogram illustrated in FIG. 2 is implemented by the system 100 previously described and comprises a first step 210 wherein the assignment module 110 receives input from a user specifying the existing planogram 130 and the new assortment of products 120.

[0075] Next step 215 consists of executing the first computer program 116A by the central processing unit 112 for storing the existing planogram 130 and the new assortment 120 in the dedicated zone 118 of the memory 114.

[0076] During a step 220, the second computer program 116B is executed by the central processing unit 112 for generating a set of at least one segmentation tree which describes the relative geometrical placement of products in the existing planogram 130 according to said ordered category types, for example “segment” and “brand” types. The root node of a segmentation tree identifies it. Then, each level of the identified segmentation tree corresponds to a category type, in the order defined by the ordered category types. In each level of the segmentation tree, each node corresponds to a possible category value of the category type which corresponds to this level. For example, in the case of two category types “segment” and “brand” where “segment” < “brand”, a segmentation tree contains three levels. First level L0 contains the root node. Second level L1 contains nodes which correspond to possible values of category type “segment”. Third level L2, which is the last level, contains leaf nodes which correspond to possible values of category type “brand”. A node in the segmentation tree may contain information on category values together with geometrical information. These informations may be recorded, for instance, through a set of products as found in the existing planogram 130. Details and result of this step execution will be described with reference to FIGS. 5 and 6.

[0077] During an optional step 230, the output planogram 140 is created by duplicating the existing planogram 130.

[0078] Next step 240 consists of executing the third computer program 116C by the central processing unit 112 for removing, from the output planogram 140, products which are in the existing assortment of the existing planogram 130 but not in the new assortment 120. If the number of occurrences of any product P in the new assortment 120, denoted

$n_s(P)$, is known and also the number of occurrences of any product P in the existing assortment of the existing planogram 130, denoted $n_e(P)$, then when a product P is present in both the new assortment 120 and in the existing assortment of the existing planogram 130 but such that $n_s(P) < n_e(P)$, $n_e(P) - n_s(P)$ occurrences of P are removed from the output planogram 140. The corresponding length of linear thus removed is kept in memory, where the length of linear for a set of products is defined as the sum of their widths.

[0079] Next step 250 consists of executing the fourth computer program 116D by the central processing unit 112 for adding, to the output planogram 140, products which are in the new assortment 120 but not in the existing assortment of the existing planogram 130, wherein adding the new products to the output planogram 140 uses the set of at least one segmentation tree generated at step 220. Details of this step execution will be described with reference to FIGS. 7, 8 and 9.

[0080] Final step 260 consists of executing the fifth computer program 116E by the central processing unit 112 for storing the output planogram 140 in the dedicated zone 118 of the memory 114.

[0081] Step 230 is optional because, in another embodiment, the output planogram 140 can be the existing planogram 130 from which products which are in the existing assortment but not in the new assortment 120 are directly removed and to which products which are in the new assortment 120 but not in the existing assortment are directly added. In that case, it means that the output planogram 140 is created by modifying the input planogram 130 in-place.

[0082] In another embodiment also, some of the previously detailed steps may be followed in a different order, without diverting from the scope of the present invention. For instance, in a possible embodiment, removing products at step 240 could follow the addition of products at step 250. In another possible embodiment, where step 230 is deleted and step 240 follows step 250, steps 220 and 250 can be processed together, so that segmentation trees are created as needed when adding products to the input/output planogram 130/140.

[0083] An example of new assortment 120 is shown in FIG. 3. This new assortment 120 is a finite set of products. Each product P of this finite set, identified by an identification number (such as “145”, “223”, . . .), is characterized by a number n of category values, $\{c_1(P), \dots, c_n(P)\} \in C^n$, wherein each category value $c_i(P)$ is selected from a finite set of possible values in a corresponding category type c_i . In the example of FIG. 3, two category types are defined ($n=2$), c_1 =“Segment” and c_2 =“Brand” as mentioned previously. The category type c_1 =“Segment” is associated to four possible category values S1, S2, S3 and S4 while the category type c_2 =“Brand” is associated to four other category values A, B, C and D.

[0084] Optionally, variables may be defined to denote the number of times a product P appears. In that case, the number of times a product P appears in the new assortment 120 is denoted by $n_a(P)$ as mentioned previously.

[0085] Furthermore, each product P in the new assortment 120 has so-called geometrical properties, typically a product width.

[0086] For instance, for product P identified by identification number 145: $c_1(P=145)=S1$, $c_2(P=145)=A$ and

$n_a(P=145)=4$; for product P identified by identification number 223: $c_1(P=223)=S1$, $c_1(P=223)=B$ and $n_a(P=223)=1$; etc.

[0087] An example of existing planogram 130 is shown in FIG. 4. This existing planogram 130 includes an existing assortment of products in association with a geometrical placement layout of these products on shelves. In this existing assortment, $n_p(P)$ denotes the number of occurrences of product P. S1, S2, S3, S4 and S5 are five possible values of category type c_1 ="Segment". They indicate five different areas in the illustrated set of shelves delimited by dotted lines. A, B, C and D are the four possible values of category type c_2 ="Brand" to which each product P of the existing assortment can be associated. In this example, category types c_1 and c_2 are ordered such that $c_1 < c_2$. In more general terms, category types are ordered according to the invention such that one of the category type corresponds to each category value $C \in C$, noted $t(c)$, and for each pair of category values $(c_i, c_j) \in C^2$, it is true that either $t(c_i)=t(c_j)$ or $t(c_i) > t(c_j)$, there exists an ordering defined on the set of category values C.

[0088] In addition to these values in each category type for each product P (i.e. each product P is located in one of the segments S1, S2, S3, S4, S5 and belongs to one of the brands A, B, C, D), and in addition to product P specific geometrical properties such as a width, each product P present in the existing planogram 130 has geometrical properties describing its position in the existing planogram 130 (i.e. placement data): such geometrical properties are for instance the shelf on which it is placed, and its position on this shelf.

[0089] Optionally, the existing planogram 130 may be partitioned into K groups where a group is a set of shelves at the same height. An example group G_k of shelves at the same height is shown in FIG. 4. A segmentation tree is then created for each group G_k , based on the physical position of products on the shelves of this group and on their category values, as it will now be shown in reference with FIG. 5. K segmentation trees are thus created.

[0090] FIG. 5 illustrates details of step 220 of the method in FIG. 2 for generating a set of at least one segmentation tree from the existing planogram 130. In the example of FIG. 3, K segmentation trees are generated, one for each group G_k .

[0091] In the case of n category types $c_1, \dots, c_n$, let us consider that these category types are ordered such that $c_1 < \dots < c_n$. Each node N at level L_m of a segmentation tree, where $m \leq n$, corresponds by inheritance from the root node to a list of ordered category values $C(N) = \{c_1(P), \dots, c_m(P)\}$. As a consequence, if N is a child of a node M, then $C(M) \subset C(N)$ and C(M) has one item less (i.e. item $c_m(P)$) than C(N).

[0092] At a first step 510, the existing planogram 130 is partitioned into K groups G_k , $1 \leq k \leq K$.

[0093] Next, at step 520, for each group G_k , a root node N0 is created which identifies this group and its corresponding segmentation tree $T(G_k)$. Then, the segmentation tree $T(G_k)$ is generated by iterating over the products of the existing planogram 130 in group G_k .

[0094] Therefore, at following step 530, a test is performed to get next product P in group G_k . If there is no more product left in group G_k , then stop adding nodes to tree $T(G_k)$ at step 540 and get next group.

[0095] If there is at least one product left in group G_k , then get next product P and a test is performed at step 550 for each category type $c_1, \dots, c_n$ to see whether product P can be associated to at least one existing node N in the segmentation

tree $T(G_k)$. The test consists of looking for a node N at any level L_i such that $\{c_1(P), \dots, c_i(P)\} \subset C(N)$.

[0096] If there exists no node N at any level L_i such that $\{c_1(P), \dots, c_i(P)\} \subset C(N)$, then go to step 560 for creating such a new node with all necessary ancestors from the root node to complete the corresponding branch of the segmentation tree.

[0097] If there already exists a node N at any level L_i such that $\{c_1(P), \dots, c_i(P)\} \subset C(N)$, or following step 560, go to step 570 for updating geometrical information associated to said node N and its ancestors. At step 570, placement data, such as the identification of the shelf on which the product P is placed, its position on this shelf and the length of linear it occupies, are used for updating such geometrical information associated to node N. Following step 570, go to step 530.

[0098] In a possible embodiment of step 220, the ordering of category types $c_1, \dots, c_n$ may be fixed. In another possible embodiment, the ordering of category types may be decided by the user and given as input to the method. Yet in another possible embodiment, the ordering of category types may be dynamically adapted to the structure of the existing planogram 130.

[0099] As illustrated in FIG. 6, in the case of group G_k surrounded by dashed lines of the existing planogram 130 as illustrated in FIG. 4, and assuming that "segment" type < "brand" type, a segmentation tree with three levels L_0 (for root node N0), L_1 (for nodes corresponding to possible values of "segment" type) and L_2 (for leaf nodes corresponding to possible values of "brand" type) is generated at step 220 while executing steps 510 to 570. In this segmentation tree, there is no node associated to segment S4 because segment S4 is absent from surrounded group G_k in FIG. 4. There is no child node of node N1 associated to brand D because brand D is absent from segment S1 in surrounded group G_k . There are no child nodes of node N2 associated to brands A, B and D because brands A, B and D are absent from segment S2 in surrounded group G_k . There are no child nodes of node N3 associated to brands B, C and D because brands B, C and D are absent from segment S3 in surrounded group G_k . And there are no child nodes of node N4 associated to brands A and D because brands A and D are absent from segment S5 in surrounded group G_k .

[0100] FIG. 7 illustrates details of step 250 of the method in FIG. 2 for adding, to the output planogram 140, products which are in the new assortment 120 but not in the existing assortment of the existing planogram 130. The positions where the new products are added to the output planogram 140 are derived from the segmentation trees generated at step 220.

[0101] At a first step 710, a subset A of products in the new assortment 120 but not in the existing assortment of existing planogram 130 is created, which is the subset of products in the new assortment 120 to add to the output planogram 140. Each product of subset A is associated to all nodes of any segmentation tree with compatible category values. In particular, it can happen that two different nodes N1 and N2 from two different segmentation trees share the same set of category values, i.e. $C(N1)=C(N2)$. In that case, these nodes N1 and N2 are said to be "similar" and are both associated to the same products of subset A. For a product P in the new assortment 120, if $n_a(P)$ is defined and $n_a(P) > n_p(P)$, then product P is in subset A. If $n_a(P)$ is not defined, the same rule can be used

to build subset A, with n_a defined as $n_a(P)=n_p(P)$ if $n_p(P)>0$ and $n_a(P)=1$ otherwise. Or $n_a(P)$ is kept undefined and then P is in subset A only if $n_p(P)=0$.

[0102] In this first step 710, for any node N in the set of segmentation trees, a list of products from subset A with compatible categories, denoted $P(N) \subseteq A$, is created.

[0103] Let $C(P)=\{c_1(P), \dots, c_n(P)\} \in C''$ be the set of ordered category values of a product P. For any product P of subset A, two cases may arise:

[0104] a) there is at least one node N in the set of segmentation trees such that $C(P)=C(N)$; then $P(N)$ is updated such that $P(N) \leftarrow P(N) \cup \{P\}$, or

[0105] b) there is no node N in the set of segmentation trees such that $C(P)=C(N)$; then, a "best" node is found, i.e. the node N of highest level such that $C(N) \subset C(P)$ and $P(N)$ is updated such that $P(N) \leftarrow P(N) \cup \{P\}$.

[0106] Then for all nodes N' similar to N and all their ancestors, product P is added to their lists of compatible products. Formally, we can write that $\forall M$ such that $C(M) \subseteq C(N)$, $P(M) \leftarrow P(M) \cup \{P\}$.

[0107] The segmentation trees may, or may not, be updated to add new nodes in case (b).

[0108] At a second step 720 executed for each segmentation tree, the geometric constraints of all nodes are adjusted so that room is available for all products. For example, it may consist of making sure that for each node, the length of linear available is bigger than or equal to the added length of linear available for all its children nodes. In another possible embodiment, this step may be discarded, i.e. it is assumed that all products will fit in the output planogram 140.

[0109] At a third step 730, sets of products linked with several similar nodes are partitioned. It is an optional step which is used for optimization and it may be discarded in another embodiment. Example embodiments for this step will be explained with reference to FIG. 8 wherein an example of similar subtrees is illustrated.

[0110] Finally, at a fourth step 740, products of subset A are added iteratively to the output planogram 140 at most until there is not room enough to add more occurrences of any product. If the number of occurrences $n_s(P)$ is defined for each product P in the new assortment 120, then products of subset A are added until one of the two following conditions is met for each product P:

[0111] the number of occurrences $n_a(P)$ has been reached in the output planogram 140,

[0112] there is not room enough to add more occurrences of this product.

[0113] In a possible embodiment, at step 740, the condition that there is enough room to add a product may be ignored in the case where a product must be added that is not present in the planogram ($n_p(P)=0$). In this case, at least one facing is added for each product in the new assortment 120 and possible geometrical constraints would then be enforced as a final step.

[0114] An example of iterative addition of products from subset A to the output planogram 140 will be detailed with reference to FIG. 9.

[0115] As above mentioned, FIG. 8 shows two subtrees from two different segmentation trees with three similar nodes N0, N1, N2 and N'0, N'1, N'2 respectively, i.e. $C(N0)=C(N'0)$, $C(N1)=C(N'1)$ and $C(N2)=C(N'2)$. FIG. 8 further shows a set of products P1, ..., P7 that have been associated with each node. Products P1 to P7 are associated with parent nodes N0 and N'0; products P1, P2 with children nodes N1

and N'1; and products P3, P4, P5 with children nodes N2 and N'2 (shown only for the first subtree in the left part of FIG. 8). Two products P6 and P7 are associated with the parent nodes N0 and N'0 but with none of the children nodes. At step 730, the length of linear available for node N0 (respectively N'0) should be strictly bigger than the added lengths of its children N1 (respectively N'1) and N2 (respectively N'2), in order to accommodate for products P6 and P7.

[0116] As an example of partition executed at step 730, in the example of FIG. 8, there are two similar nodes N1 and N'1, which consequently have the same set of associated products $P(N1)=P(N'1)=\{P1, P2\}$. Then it is possible to partition this set so that P1 will be added to N1 only, and P2 to N'1 only, i.e. $P'(N1)=\{P1\}$ and $P'(N'1)=\{P2\}$. The choice here is arbitrary, but in another embodiment it could be optimized using geometrical information. For nodes N2 and N'2, as there are three products to partition, it may be useful to use geometrical information from the nodes together with the widths of the products to optimize the partition.

[0117] As another example of partition executed at step 730, in the example of FIG. 8, let's take the nodes N2 and N'2, with available linear lengths of 60 and 20 respectively. Let $P(N2)=P(N'2)=\{P3, P4, P5\}$. Each product P has a certain width $w(P)$: e.g. $w(P3)=10$, $w(P4)=5$, $w(P5)=5$. Suppose $n_a(P)$ is not defined for all products. Total product length with one occurrence for each product is 20. An optimum partition would give 75% of the product widths attached to node N2, and the remaining 25% attached to node N'2, following the proportions of available linear lengths. A best partition is found, e.g. $P'(N2)=\{P3, P4\}$ (75%) with node N2, and $P'(N'2)=\{P5\}$ (25%) with node N'2. With this partition, it is possible to add four occurrences of each product. Note however that this is not a partition in the mathematical sense, in that a product may still be assigned to several nodes.

[0118] The iterative addition of products (step 740) from subset A to the output planogram 140 as illustrated in FIG. 9 assumes that the number of occurrences $n_a(P)$ of each product P in the new assortment 120 is known.

[0119] At a first step 910, a product count $m(P)=n_a(P)$ is defined and initialized for each product P in the subset A.

[0120] At next step 912, a test is performed to get next product P in subset A. If there is no more product left in subset A, then stop adding products at final step 914.

[0121] Following step 912, once next product P has been selected from subset A, a test 916 is executed to see whether a next best node N for this product P is available, i.e. the next best node N such that product P belongs to $P(N)$.

[0122] Following step 916, if no node is available for adding product P, then go to step 918 for stopping adding this product P and then go to step 912 for getting next product in subset A.

[0123] Following step 916, if a node N is available for adding product P, then go to step 920 for testing the value of product count $m(P)$. Test 920 consists of determining whether $m(P)$ is positive or null.

[0124] Following step 920, if $m(P)$ is null, then go to step 918 for stopping adding this product P and then go to step 912 for getting next product in subset A.

[0125] Following step 920, if $m(P)$ is positive, then go to step 922 for testing whether it is possible to add product P to the output planogram 140 at node N. It is possible to add product P for example if there is still some linear space available as conditioned by node N.

[0126] Following step 922, if it is not possible to add product P, then go to step 916 to determine whether another node is available for adding product P to output planogram 140.

[0127] Following step 922, if it is possible to add product P to output planogram 140 at node N, then go to step 924. At this step, use the geometrical information of node N to place the product P on a corresponding shelf, further using any optimization criteria for placement and number of facings of products on the shelves. Update the space available accordingly in node N geometrical information at next step 926.

[0128] At next step 928, decrease product count for product P by one (i.e. $m(P) \leftarrow m(P) - 1$) and go to step 912 for getting next product in subset A.

[0129] In case the number of occurrences $n_a(P)$ of each product P in the new assortment 120 is unknown, then the iterative addition (step 740) of products from subset A to the output planogram 140 may start directly at step 912. In that case too, steps 920 and 928 can be removed and step 912 can follow directly step 926.

[0130] Optionally, when placing a new product in step 740, priority may be given to nodes from lower levels when enforcing geometrical constraints. In a possible embodiment, relative positions may be used to place all the new products, moving other products already in place if necessary to avoid collisions, and a finer positioning algorithm may be used to rearrange the products where a node's geometrical constraints are violated.

[0131] Optionally also, in a possible embodiment, the method for adding products at step 740 may be simplified to add all items of a product at once, in case there is only one node available for a product, or by batches, in case there are several nodes available.

[0132] Optionally also, in another embodiment, the products may be ordered before step 740, so that products are added first to the nodes with fewest products.

[0133] It is clear that a method such as the one described previously, implemented in the system described previously, makes it possible to efficiently generate an output planogram that is suited to the specific needs of a retail store.

[0134] It will also be noted that the invention is not limited to the embodiments described previously. It will be clear to a person skilled in the art that various modifications can be made to the embodiments described above in the light of the teaching that has just been disclosed to him. In the following claims, the terms used must not be interpreted as limiting the claims to the embodiments disclosed in the present description, but must be interpreted so as to include therein all the equivalents that the claims aim to cover because of their wording and the prediction of which is within the capability of a person skilled in the art applying his general knowledge to the use of the teaching that has just been disclosed to him.

1. A method for assigning an assortment of products to an existing planogram comprising the following steps:

- providing with an existing planogram which includes an existing assortment of products in association with a geometrical placement layout of these products on shelves and storing said existing planogram in memory,
- providing with a user-defined new assortment of products and storing said new assortment in memory,
- furnishing, by a processor which has read/write access to the memory, an output planogram by:
 - removing, from the output planogram, products which are in the existing assortment but not in the new assortment,

- adding, to the output planogram, products which are in the new assortment but not in the existing assortment, storing said output planogram in memory, wherein the method further comprises:

- generating, by the processor, a set of at least one segmentation tree which describes the relative geometrical placement of products in the existing planogram according to several ordered category types, each category type corresponding to a level of said at least one segmentation tree,

- adding the new products to the output planogram using the generated set of at least one segmentation tree.

2. The method for assigning an assortment of products to an existing planogram as claimed in claim 1, wherein adding new products to the output planogram using the generated set of at least one segmentation tree further uses optimization criteria for placement and number of facings of products on the shelves.

3. The method for assigning an assortment of products to an existing planogram as claimed in claim 1, wherein the ordered category types include at least a "segment" type which defines several geometrically segmented areas of the shelves and a "brand" type which defines several brands to which every product can be associated.

4. The method for assigning an assortment of products to an existing planogram as claimed in claim 1, wherein the output planogram is the existing planogram from which products which are in the existing assortment but not in the new assortment are directly removed and to which products which are in the new assortment but not in the existing assortment are directly added.

5. The method for assigning an assortment of products to an existing planogram as claimed in claim 1, wherein generating said at least one segmentation tree includes:

- partitioning the existing planogram into several groups, wherein each group contains one or more shelves located at the same height in the existing planogram,
- generating one segmentation tree for each group.

6. The method for assigning an assortment of products to an existing planogram as claimed in claim 1, wherein generating a segmentation tree includes creating a root node that identifies this segmentation tree and iterating over the corresponding products of the existing planogram as follows:

- for each product, determine the corresponding category values taken for each category type, and

- if there exists no node in the segmentation tree at any level such that this node and its ancestors are associated to at least part of the category values determined for the product, then create such a new node with all necessary ancestors from the root node to complete a corresponding branch of the segmentation tree and update geometrical information associated to said new node and its ancestors,

- else, update geometrical information associated to said associated node and its ancestors.

7. The method for assigning an assortment of products to an existing planogram as claimed in claim 6, wherein geometrical information is updated using placement data of the product, such as the identification of the shelf on which the product is placed, the position of the product on this shelf and the length of linear the product occupies.

8. The method for assigning an assortment of products to an existing planogram as claimed in claim 6, wherein adding

the new products to the output planogram using the generated set of at least one segmentation tree includes:

creating a subset of products in the new assortment but not in the existing assortment of existing planogram, associating each product of the subset to all nodes of any segmentation tree with compatible category values, and iteratively adding each product of the subset to the output planogram in association with an available node and updating geometrical information associated with said available node.

9. A system for assigning an assortment of products to an existing planogram comprising:

a memory for storing:

an existing planogram which includes an existing assortment of products in association with a geometrical placement layout of these products on shelves, a user-defined new assortment of products, and an output planogram;

a processor which has read/write access to the memory for furnishing the output planogram by:

removing, from the output planogram, products which are in the existing assortment but not in the new assortment, adding, to the output planogram, products which are in the new assortment but not in the existing assortment, wherein the processor has read/write access to the memory for further:

generating a set of at least one segmentation tree which describes the relative geometrical placement of products in the existing planogram according to several ordered category types, each category type corresponding to a level of said at least one segmentation tree, adding the new products to the output planogram using the generated set of at least one segmentation tree.

10. A computer program that can be downloaded from a communication network and/or saved on a computer-readable medium and/or executed by a processor, said program comprising program code instructions for the execution of the steps of a method for assigning an assortment of products to an existing planogram as claimed in claim 1 when said program is executed on a computer.

* * * * *