



- (51) International Patent Classification:
G06F 17/50 (2006.01)
- (21) International Application Number:
PCT/US2012/038582
- (22) International Filing Date:
18 May 2012 (18.05.2012)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
61/495,980 11 June 2011 (11.06.2011) US
- (71) Applicant (for all designated States except US): **DIRTT ENVIRONMENTAL SOLUTIONS INC.** [US/US]; 4001 South 700 East, Suite 450, Salt Lake City, UT 84107 (US).
- (72) Inventor; and
- (75) Inventor/Applicant (for US only): **LOBERG, Barrie, A.** [CA/CA]; Site 15, Box 10 RR1, Millarville, Alberta T0L 1K0 (CA).
- (74) Agents: **FRODSHAM, Michael, J.** et al.; Workman Nydegger, 1000 Eagle Gate Tower, 60 E. South Temple, Salt Lake City, UT 84111 (US).

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

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, 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 (Rule 48.2(g))

(54) Title: AUTOMATED RE-USE OF STRUCTURAL COMPONENTS

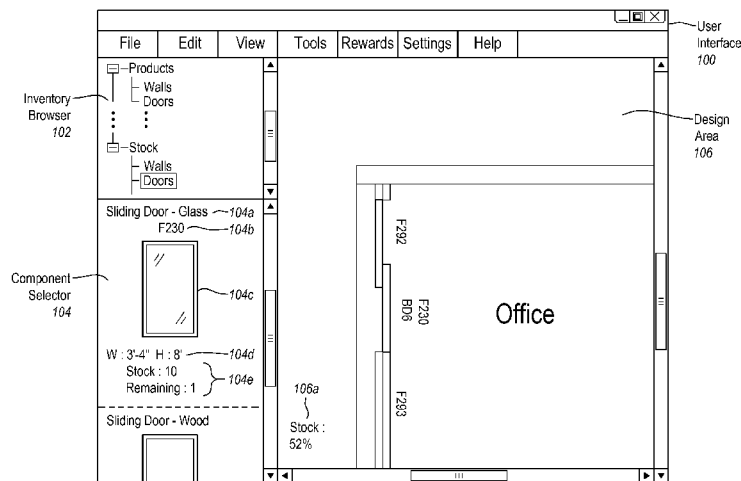


Fig. 1

(57) Abstract: Systems, methods, and apparatus allow for easy and efficient structural design and/or reconfiguration while using existing stock components. One or more implementations allow a designer to import a stock file containing an inventory of stock components that are usable as part of a structural design. In particular, one or more implementations allow the stock components to be used in a structural design while tracking, in real-time, the availability of stock components. Furthermore, one or more implementations automatically correlate stock components with existing components of a structural design or with new components as a designer inserts them into the structural design. Additionally, one or more implementations visually distinguish between new components in the structural design and stock components in the structural design. Furthermore, one or more implementations auto-select or modify surrounding components in the structural design to match an inserted stock component.

WO 2012/173741 A2

AUTOMATED RE-USE OF STRUCTURAL COMPONENTS

BACKGROUND OF THE INVENTION

Field of the Invention

5 The present invention relates generally to computer-aided design or drafting software that assists in reuse of existing stock components in a structural design.

Background and Relevant Art

 Many industries use computer-aided design (CAD) software (e.g., AutoCAD, Revit, 3DS Max, SketchUp, etc.) to design and create three-dimensional computer
10 models. Increasingly, designers use CAD software to model architectural structures. Often, designers incorporate existing structural components (e.g., modular wall panels, cubicle components, doors, HVAC components, integrated desk components, electrical components, etc.) into structural models. For example, designers may use CAD software to design a new architectural structure that makes use of existing
15 components, or may reconfigure an existing architectural structure. Designers may also use CAD software to incorporate existing non-structural components (e.g., furniture) into structural models.

 When modeling an architectural structure using existing components, designers must identify the availability of the existing stock components to develop an
20 inventory of available stock. For example, designers may physically walk through an existing structure, noting various components that make up the structure. In such situations, taking an inventory of existing components is easiest when the existing structure is already unassembled. In these situations, designers cannot incorporate existing components into new designs until after disassembly of the existing structure.
25 Designers may also consult old purchase orders, or manually enumerate a list of components used in old designs.

 In addition, designers who use conventional CAD software to incorporate existing components into a structural design often need to manually track use of existing components. Designers may track which existing components they have
30 incorporated into the structural design, and which existing components remain available for use. For example, as designers add existing components to structural designs, the designers may manually note which components they have reserved for

the designs (e.g., on paper, using a spreadsheet, etc.). As designs change and evolve, designers typically have to constantly update their notes. Doing so can be time consuming, tedious, and prone to error.

In addition to the foregoing, tracking reuse of available stock parts using
5 conventional CAD software typically requires a designer to complete a new design. Upon finishing the design, the CAD software creates a part list for the new design and compares it to a part list of available stock parts. The CAD software then provides the number of stock parts used in the new design. If un-satisfied with the amount of inventory parts being used in the new design, conventional CAD software requires the
10 designer to blindly create a new design in an attempt to see if they can incorporate more stock or inventory parts.

Furthermore, conventional CAD software provides limited information about where or how the stock parts are used in the new design. As mentioned previously, typically conventional software will only compare part lists. Thus, an installer trying
15 to build the new design using both stock parts and new parts often has to exert greater effort and time in an attempt to determine where and how the stock parts fit in the new design.

Accordingly, there are a number of problems in the art relating to reusing existing stock components in structural designs.

20 BRIEF SUMMARY OF THE INVENTION

Implementations of the present invention overcome one or more problems in the art with systems, methods, and apparatus configured to allow easy and efficient structural design and/or reconfiguration while reusing existing stock components. In particular, one or more implementations allow a designer to import a stock file
25 containing an inventory of stock components that are usable as part of a new structural design. One or more implementations allow for the use of the stock components in a new structural design while tracking, in real-time, the availability of stock components. One or more implementations automatically correlate stock components with existing components of a structural design, or with new components
30 as designers insert them into the structural design. Additionally, one or more implementations visually distinguish between new components in the structural design and stock components in the structural design.

For example a computer-implemented method for visually modeling a new structural design, in real-time, at a structural design software application can include a computer system importing a stock file. The stock file can include a plurality of existing components of an existing structural design that are available for use in the new structural design. The method can also include the computer system identifying each of the existing components that are in the stock file. In addition, the computer system can receive user input specifying a new design element of the new structural design. The computer system can automatically identify that one or more of the existing components correspond with one or more components of the new design element. The computer system can associate the automatically identified existing components with the new structural design, and visually distinguish between any existing components and any new components that are associated with the new structural design.

In addition, a method for modifying a structural design in real-time at a structural design software application can include a computer system receiving both a new structural design and a stock file. The stock file includes a plurality of existing components that are part of an existing structural design, and which are available for use in the new structural design. The computer system identifies each of the existing components that are in the stock file, and identifies existing components that are usable the new structural design. The computer system associates the identified existing components that are usable the new structural design with the new structural design.

This summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

Additional features and advantages of the invention will be set forth in the description which follows, and in part will be obvious from the description, or may be learned by the practice of the invention. The features and advantages of the invention may be realized and obtained by means of the instruments and combinations particularly pointed out in the appended claims. These and other features of the

present invention will become more fully apparent from the following description and appended claims, or may be learned by the practice of the invention as set forth hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

5 In order to describe the manner in which the above-recited and other advantages and features of the invention can be obtained, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

15 Figure 1 illustrates a schematic diagram of a user interface of an exemplary design software application with which an implementation of the present invention can integrate to use stock components in a structural design;

Figure 2 illustrates a schematic diagram of a user interface of an exemplary design software application with which an implementation of the present invention can integrate to present an analysis of use of stock components in a structural design;

20 Figure 3 illustrates an exemplary computer architecture for integrating stock components in a structural design in accordance with an implementation of the present invention;

Figure 4 illustrates a flowchart of a series of acts in a method in accordance with an implementation of the present invention of visually modeling a new structural design, in real-time, at a structural design software application; and

25 Figure 5 illustrates a flowchart of a series of acts in a method in accordance with an implementation of the present invention of modifying a structural design, in real-time, at a structural design software application.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

30 Implementations of the present invention overcome one or more problems in the art with systems, methods, and apparatus configured to allow easy and efficient structural design and/or reconfiguration while reusing existing stock components. In particular, one or more implementations allow a designer to import a stock file

containing an inventory of stock components that are usable as part of a new structural design. One or more implementations allow for the use of the stock components in a new structural design while tracking, in real-time, the availability of stock components. One or more implementations automatically correlate stock components with existing components of a structural design, or with new components as designers insert them into the structural design. Additionally, one or more implementations visually distinguish between new components in the structural design and stock components in the structural design.

In particular, one or more implementations of the present invention enable programmatic evaluation of pre-existing stock components, and automatically applies the evaluation to a new design configuration. In addition, one or more implementations enable continual optimization of the use of pre-existing stock components as the new design changes, and thus constantly re-solving a reconfiguration. One or more implementations of the present invention also provide the ability to programmatically evaluate stock components at a part or assembly level, and to provide alternative assemblies for use in the new configuration.

Such implementations can also accept user input to identify components which are damaged or otherwise not suitable for re-use in the new design, and exclude such components from the new design as applicable. In addition, one or more implementations of the present invention provide the ability to accept user input to identify components which are to be used in specific locations in the new configuration, and lock such parts or assemblies in place. Furthermore, implementations of the present invention provide the ability to generate detailed reports of pre-existing parts or assemblies to show what has been used in the new configuration, what is still left in pre-existing inventory, and what needs to be ordered to create the new configuration.

For example, at least one implementation relates to developing an inventory of available stock components (e.g., modular components, parts, assemblies, furniture). The systems, software, and methods of the present invention can import and utilize a “stock file,” which describes stock components that are already owned by an entity and that are available for use in a structural design or reconfiguration. In one or more implementations, a design software application generates the stock file based on a

previous structural design. Alternatively, the stock file may comprise a previous structural design file itself. In still further implementations, an inventory software application, a spreadsheet software application, a database software application, or any other appropriate software application generates the stock file.

5 Using the inventory of available stock components, one or more implementations automatically incorporate the stock components into a new structural design, or a reconfiguration of an existing structural design. When doing so, the systems, software, and methods can automatically update the stock inventory in real-time, and indicate to the designer which stock components have been used and/or
10 which stock components remain. Furthermore, one or more implementations may apply stock components to a structural design as the designer creates the design (i.e., as the designer inserts design elements into the structural design). Additionally or alternatively, implementations of the present invention may apply stock components to an existing structural design (i.e., by replacing components of a structural design
15 with stock components).

 In one or more implementations, systems, software, and methods of the present invention visually distinguish between stock components in a structural design and new components in the structural design. For example, a design software application may apply one color to stock components and a different color to new
20 components. In another example, a design software application may apply different colors to component identifiers. In yet other examples, design software applications may apply different textures and/or shading to components to distinguish between stock components and new components.

 In one or more implementations, systems, software, and methods of the present invention distinguish between stock components in a structural design and
25 new components in the structural design using text. For example, a design software application may annotate stock components and new components as such using text labels (e.g., “new” or “stock”). A design software application may also append or prepend text labels to component identifiers to identify whether a component is new
30 or stock.

 Thus, one or more implementations allow the designer or user to note where each stock part or assembly came from and how many more remain in inventory.

Furthermore, all information is updated graphically and instantaneously. Thus, one or more implementations can help reduce design time, and increase the reuse of parts and assemblies. The real-time visual/graphical feedback and lead to a reduction of unnecessarily ordering new product. One will appreciate in light of the disclosure
5 herein that one or more implementations can reduce the expensive and damaging mindset of disposing viable materials and products.

In addition, one or more implementations of systems, software, and methods of the present invention adjust a structural design to accommodate stock components and/or to maximize re-use of stock components. For example, in response to a
10 designer inserting a component, a design software application may automatically adjust surrounding components to accommodate the inserted component. Accommodating an inserted component may include changing surrounding components to different types, size, or style of new components or to different stock components.

One will thus appreciate that because the systems, methods, and computer
15 program products of the present invention include design components that include associated product attributes and parameters, a non-expert can easily and efficiently create accurate, designs incorporating existing or stock parts and/or assemblies. Furthermore, implementations of the present invention can apply rules and behaviors
20 and interpret design intent automatically, not only providing an accurate layout, but also allowing change at any time without rework. Implementations of the present invention can also recognize required changes effected by user driven changes (i.e., choosing to replace a given design component with a stock component), and automatically resolve the design layout, including required component substitutions
25 and dimensional changes, to help maintain validity.

Along similar lines, one or more implementations of the present invention provide software that can recognize required changes affected by the user-driven change, and re-solve the layout design to automatically include required component additions, component substitutions, and/or dimensional changes effected by the user
30 driven design change. The design software system can change product types (e.g., from new components to stock components, with its own set of independent behaviors and product rules), and resolve to the design and its relative behaviors and rules. An

auto-solve function in the design software system can identify areas of conflict in the design and either solve the conflicts automatically, or provide choices for the correction. The design software system can also implement the selected solution automatically (e.g., once the end-user has made a choice).

5 Generally, a user can create layouts in a two-dimensional plan view, although users need not work in a linear sequential manner. In most cases, the user is free to design in any manner they wish, and the design software system will understand the design intent and resolve the design accordingly. Additionally, implementations of the present invention can represent the design space in three-dimensional graphics.
10 Thus, a user can easily and quickly toggle between different views or representations to gain a complete understanding of the design space and design components therein. The design software system can create each of these separate views using the same dataset. Thus, one will appreciate that implementations of the present invention can eliminate the need to audit between information sources. Furthermore, because each
15 representation can be based on the same dataset, the design software system can instantly and simultaneously resolve and modify all representations.

 Furthermore, through the use of an object-oriented programming language (e.g., Java), the design software system can create programming objects for each component, attribute, option, relationship, and behavior. The design software system
20 can store these objects within a single data structure. Additionally, the design software system can cause each object, in turn, to resolve itself in the design. The design software system can thus appropriately propagate user input at any level of this dataset to all objects in the data structure, thereby resolving the complete design through the resolving of each object individually. In addition, implementations of the
25 present invention can include a solution level programming object, which recognizes behaviors and relationships between objects within the design. As also discussed above, the solution-level programming object can also resolve itself with each user input.

 In at least one implementation, the systems, methods, and computer products
30 of the present invention allow a user to select various design components from a product menu, which the user can then add to a design space. One will appreciate that a user can also select the design components from a product menu, or the user can

import them into the design software system from another software program. For example, Figure 1 depicts a schematic diagram of a user interface 100 of an exemplary design software application with which implementations of the present invention can integrate to reuse stock components in structural design. The user interface 100 enables designers to incorporate components from a stock inventory into a structural design. The user interface 100 can enable designers to use stock components exclusively in structural designs, or can enable designers to combine stock components with new components (i.e., components to purchase) in structural designs. As depicted, the user interface 100 includes an inventory browser 102, a component selector 104, and a design area 106.

The inventory browser 102 presents a selection of components that are available for use in structural designs. As depicted, for example, the inventory browser 102 may present a hierarchical listing of component categories and sub-categories, though any appropriate selection mechanism falls within the scope of the present invention. The inventory browser 102 can enable designers to select both new components and stock components for use in structural designs. As depicted, for example, the inventory browser 102 includes a “products” category that contains sub-categories (e.g., walls, doors) of new components, and a “stock” category that contains sub-categories (e.g., walls, doors) of stock components. As used herein the term “stock” refers to a part, assembly, or subassembly which has previously been purchased and is either currently in use in a space or structure that is being redesigned, or is otherwise available in inventory for use. A “new” product on the other hand refers to a part, assembly, or subassembly that is not in inventory and needs to be purchased.

The component selector 104 presents components that are available within a particular selection of the inventory browser 102. For example, when a designer selects a category or sub-category in the inventory browser 102, the component selector 104 may present components within the selected category or sub-category. As such, depending on the category selected in the inventory browser 102, the component selector 104 may present either new components or stock components. For example, if a designer selects the depicted “doors” sub-category of the “products” category, then the component selector 104 may display new door components that are

available for purchase and integration into a structural design. Similarly, if the designer selects the depicted “doors” sub-category of the “stock” category, then the component selector 104 may display components from the stock inventory, which are available for integration into the structural design.

5 In the depicted example, a designer has selected the “doors” sub-category of the “stock” category. Thus, the component selector 104 displays stock door components that are available for use in a structural design. The component selector 104 may present components in any appropriate manner. As depicted, for example, the component selector 104 may display, for each component, one or more of a title
10 104a, a part number 104b, a visual representation 104c (e.g., a photograph or sketch), or product specifications (e.g., dimensions, style, etc.). When presenting stock components, component selector 104 may present stock information 104e. Stock information 104e can present information about existing stock of the item, such as the total number of the component that are available in stock, that are used in the current
15 structural design, that remain available for use, that are damaged, etc. Component selector 104 may present other information, such as price, estimated shipping price, estimated shipping time, etc.

Components can become available in the inventory browser 102 and the component selector 104 through integration into the design software application, or
20 through importation. For example, the design software application may include an “import” function that enables users to import stock files, libraries, etc. For example, the user interface 100 may include an import menu selection, such as File > Open, File > Import, etc., though any appropriate import selection falls within the scope of the present invention. When a user loads a stock file using the import function, the
25 design software application creates a library of available styles in the inventory browser 102 / component selector 104 based on styles contained in the stock file. In one or more implementations, the design software application creates the library regardless of whether or not an existing structural design is loaded in the design software application.

30 In one or more implementations, one or both of the component selector 104 or the inventory browser 102 distinguish between components and assemblies. In general, components represent the smallest discrete unit represented in user interface

100, and assemblies represent combinations of components. For example, a door assembly may include as components a door, a door frame, and door hardware (e.g., a knob, a handle, hinges, etc.). As such, designers may use the component selector 104 and/or the inventory browser 102 to browse and select items at both the assembly
5 level and the sub-assembly (i.e., component) level. In one or more implementations, assemblies may include other assemblies, such as a room assembly that includes wall assemblies and a door assembly. Providing both assemblies and components affords designers the simplicity and convenience of working with assemblies, as well as the flexibility and customization available through use of components. In one or more
10 implementations, the design software application enables designers to create and use their own assemblies.

The design area 106 visually presents structural designs, and enables designers to add or remove components or assemblies with respect to structural designs. As part of the presented structural designs, the design area 106 can visually present
15 various components or assemblies, as well identifiers (e.g., part name, part number, etc). In the depicted example, the design area 106 presents a portion of an office, including component identifiers. For example, the depicted design area 106 identifies two wall assemblies (with part identifiers F292 and F293), a sliding glass door (part number F230) and a door frame (part number BD6).

20 In one or more implementations, the design area 106 visually distinguishes between new components (or assemblies) and stock components (or assemblies). For example, the design area 106 may present stock components in one color (or one set of colors), and present new components in a different color (or different set of colors). Additionally or alternatively, the design area 106 may apply a first shading (or first
25 set of shadings) to stock components, and apply a different shading (or different set of shadings) to new components. In one or more implementations, the design area 106 may present component identifiers in different colors to indicate which components are from stock, and which components are new. Of course, any other mechanism for visually distinguishing between stock components and new components falls within
30 the scope of the present invention.

One will appreciate in light of the disclosure herein that the visual distinction of stock component from new components can provide the designer with important

feedback. For example, a designer may note that a portion of the design (i.e., an assembly such as a room or wall) includes only new parts. The designer can then review available stock components using the inventory browser 102 to find stock components to replace the new components. Furthermore, the designer can readily
5 note exactly where in the design stock components are being used.

One or more implementations of the design software application are configured both to edit existing structural designs and to create new structural designs. In either case, the design software application makes it easy to incorporate existing stock into structural designs. As indicated previously, one or more
10 implementations of the user interface 100 provide import/open functionality that enables importation of stock files. Stock files may comprise existing designs (e.g., the design in which the stock components are presently arranged), inventories generated from existing designs, spreadsheets, and the like. When the design software application receives a stock file, the design software application identifies
15 components included in the stock file and adds them to the stock inventory. The design software application then makes these stock components available to designers for use in structural designs.

In one or more implementations, the user interface 100 is configured to enable a designer to provide existing structural designs to the design software application.
20 For example, the user interface 100 may provide import/open functionality that enables importation of existing structural designs. When the design software application receives an existing structural design the design software application can be configured to compare components of the existing design with available stock. The design software application can also be configured to automatically substitute
25 matches between components and/or assemblies of the existing design with components and/or assemblies from the available stock. For example, when the design software application receives an existing design and a stock file, the design software application can automatically substitute stock component with components in the existing design. When doing so, the design software application can substitute
30 components at the assembly level and/or at the sub-assembly level. The user interface 100 can present the updated design in the design area 106 while visually

distinguishing between stock components and new components, and designer can use user interface 100 to modify the imported design, if desired.

In one or more implementations, the user interface 100 is configured to enable a designer to create a structural design from scratch, or to manually modify a structural design. For example, a designer may select a component from the component selector 104, and place the component within the design area 106 (e.g., via drag-and-drop or any other appropriate operation). Depending on the type of component the designer is placing, the design software application can take one or more actions. If the component the designer is placing is from stock, the design software application may place the component in the design, visually indicate that the component is from stock, and automatically update the stock inventory to reflect that the component is part of the design. For example, if the component the designer is placing is the sliding glass door depicted in connection with component selector 104, the design area 106 may visually distinguish the component itself and/or its designator (F230) as being from stock. In addition, the component selector 104 may automatically update the stock information 104e to reflect use of the component.

Alternatively, if the component being placed in the new design is a new (i.e., non-stock) component, the design software application can compare the new component to the stock inventory. If the design software application finds a match between the new component and a component in the stock inventory, the design software application can associate the matching stock component with the new design. In other words, the design software application can place the matching stock component in the design in place of the new component. The design software application may also visually indicate that the component is from stock, and automatically update the stock inventory to reflect that the component is part of the design as part of associating the matching stock component with the new design. For example, if the component the designer is placing is a new sliding glass door that matches the stock sliding glass door depicted in connection with component selector 104, the design software application may place the stock door in the design instead of the new door. The design software application may also update the design area 106 to visually distinguish the component and/or its designator (F230) as being from stock,

and update the component selector 104 to display updated stock information 104e that reflects use of the stock component.

In one or more implementations, the design software application automatically substitutes new components or assemblies with their stock counterparts without user interaction. In additional or alternative implementations, the design software application automatically matches the components or assemblies, but prompts the designer prior to making a substitution.

When matching components, the design software application may match based on one of several parameters, such as part type and visual style. In general, different components may have the same type or functionality (e.g., size, compatibility with adjacent structure or parts), but may have different visual styles. For example, different doors of the same functional type may comprise finishes of different colors, or may comprise construction from different materials. As such, the design software application can match components based on type, or based on both type and style. Generally, when the design software application makes automatic substitutions, the design software application substitutes stock items that have an “exact” match (i.e., a match in both type and style) with the desired new item. If the design software application finds a stock component that has a “close” or “similar” match (i.e., a match in type, but not style/aesthetics), the design software application may prompt the user to place the stock component in the design, even though it is of a different style.

In one or more implementations, the design software application is configured to modify surrounding components in the design. For example, a designer may place a component or assembly that does not match the style of surrounding components. In such situations, the design software application may automatically modify the surrounding components to that they match the visual style of the placed component or assembly. In addition, a designer may place a component that does not functionally match with surrounding components. In such situations, the design software application may automatically modify surrounding components so that they match the functionality of the placed component or assembly. For example, if a designer places a sliding door in a space reserved for a swinging door, the design software application may automatically modify the door frame and surrounding walls

to accommodate the swinging door. In some situations, the design software application may modify surrounding components so that they match the size of the placed component or assembly.

In either situation, the design software application may consider both stock
5 components and new components when modifying surrounding components. For example, a designer may place a stock sliding door in a design, and the design software application may be able to modify a new surrounding door frame and new walls to accommodate the stock sliding door. When doing so, the design software application may replace the new door frame and the new wall components with a
10 stock door frame and/or stock wall components. The design software application may replace surrounding components with, or without, additional user input depending on implementation and/or user preference.

In one or more implementations, the design software application may “deconstruct” assemblies to optimize use of stock components. For example, a
15 designer may place a new door assembly that includes a door frame, a door, hinges, and a door knob. The design software application may determine that the stock inventory contains a door that could be part of the assembly. The design software application may then replace the new door component in the assembly with the stock door component, but maintain the other components as new components.

20 Additionally, the user interface 100 optionally presents a percentage of stock components used as part of a structural design. As depicted, for example, design area 106 may include a percentage 106a of stock components used in the displayed structural design. The percentage 106a of stock components used in the design can update in real time. Thus, a designer can have real time knowledge of the amount of
25 stock components being used in the design. This can allow the designer to maximize or otherwise control the percentage of stock components as the design progresses.

In one or more implementations, the design software application provides one or more user interfaces that enable designers to audit stock components used in structural designs and new components required for the structural designs. For
30 example, Figure 2 illustrates an exemplary user interface 200 of a design software application for presenting an analysis of use of stock components in a structural design. The user interface 200 is configured so that a designer can see—at a glance—

components that are used in a design, including which components are being re-used and which components need to be purchased.

As depicted, the user interface 200 enumerates components (and/or assemblies) used as part of a structural design. For example, the user interface 200 includes one or more rows 202 that specify components used in the structural design. As depicted, the rows 202 may specify one or both of stock components or new components, and can include a variety of data. For example, the depicted rows 202 include a component description, a tag (e.g., a part identifier), and a status (e.g., new, stock). The depicted rows 202 can also indicate how many of the components are in stock, how many of the components are in a drawing (structural design), how many of the components are available, and how many of the components will need to be purchased. Other data points are also possible, such as estimated component cost, estimated shipping cost, estimated shipping time, etc.

In addition, the depicted rows may indicate whether any of the components that are in stock are damaged or otherwise unavailable for use in a structural design. In one or more implementations, the user interface 200 enables a designer to specify the number of damaged components (if any). In additional or alternate implementations, stock files contain information about damaged components, and/or other user interfaces enable a designer to identify damaged components.

For example, row 202a identifies a component described as “Sliding Door – Glass,” which may correspond to the stock component depicted in component selector 104 of Figure 1. Row 202a identifies this component as being in stock, with a total of ten available. Row 202a also indicates that eight of the components are in the drawing, and that one of the components is damaged. As such, row 202a indicates that one component is available. In some situations, for a given component a design may use both stock and new units. Row 202b, for example, specifies that there are four components identified as “Sliding Door – Frame” in stock, and that there are eight of the frames used in the drawing. As such, row 202b specifies that four new frames are required.

In one or more implementations, the user interface 200 uses different colors to help distinguish between various components. For example, rows corresponding to stock components may appear in one color (or one set of colors) and rows

corresponding to new components may appear in a different color (or a different set of colors). Rows that correspond to both stock components and new components may blend colors, or appear in a yet different color (or a yet different set of colors). In addition, the user interface 200 may visually distinguish stock components that include damaged components in some manner, such as by highlighting the cell specifying the number of damaged units in a particular color.

Figure 2 also depicts that the user interface 200 may include additional component-related functionality. For example, a button 204 (manage stock) may provide functionality to import stock files. In addition, a button 206 (export) may provide functionality to export component lists, such as to a spreadsheet or purchase order format. Furthermore, a button 208 (report) may provide additional information about stock, about components used in a drawing, or any other appropriate reports. For example, reports may include reports related to cost analysis, re-use analysis, etc.

One will appreciate that the user interface 200 can further inform a designer what stock components have been used in the new design and what stock components remain. In addition to the foregoing, the user interface 200 can inform a designer of which parts need to be purchased. In one or more implementations, clicking on the description of a particular part will then show the exact location of the component in the design in a 2D viewer and/or 3D viewer. This can enable an installer to quickly determine how and where each of the stock components is used in the new design.

In one or more implementations, the design software application can present designers with the price ramifications of a design. For example, one or more of user interface 100 or user interface 200 can present a designer with a price quotation of components that are currently placed in a design. Stock components can be represented as having a zero price, since they are already owned. As such, the price quotation will generally be lower as a designer uses stock components in place of new components. The design software application can further present designers with a comparison on how much savings were obtained through the reuse of stock components.

Figure 3 illustrates exemplary computer architecture 300 for reusing stock components in a structural design. As depicted, the computer architecture 300 includes computer hardware, such as one or more user input devices 302 (e.g.,

keyboard, mouse, etc.), one or more display devices 304 (e.g., computer monitor(s)), one or more storage devices 306 (e.g., system memory, computer-readable media), and one or more processors 308. In addition, the computer architecture 300 includes a design software application 310. The computer architecture 300 may execute the design software application 310 using the storage device(s) 306 and the processor(s) 308. In doing so, the computer architecture 300 may provide input to the design software application 310 using user input device(s) 302 and storage device(s) 306, and may display user interfaces of the design software application 310 using the display device(s) 304.

As depicted, the design software application 310 can include a plurality of modules, such as a design module 310a, a stock inventory module 310b, an association module 310c, an import module 310d, and a user interface module 110e. One will appreciate that functionality of the depicted modules can be implemented in a variety of manners, using software modules (e.g., functions, routines, classes, objects, code files, etc.) that map directly to the depicted modules, or using software modules that do not map directly to the depicted modules. For example, a single software module may implement the functionality of multiple of the depicted modules (or even all of them). In another example, multiple software modules may be used to implement the functionality of a single one of the depicted modules.

One or more implementations of the design module 310e load, display, and provide editing functionality for structural designs. For example, the design module 310e may provide programmatic logic behind the user interfaces 100, 200 of Figures 1 and 2. One or more implementations of the design module 310e also distinguish between new components and stock components, and incorporate stock components into structural designs (either automatically or with user interaction).

One or more implementations of the stock inventory module 310b maintain an inventory of stock components. For example, the stock inventory module 310b may receive one or more stock files (e.g., from the import module 310a) and identify stock components that are included in the stock file(s). The stock inventory module 310b may also maintain one or more databases of stock components and assemblies. The stock inventory module 310b may use the storage device(s) 306 for storage of the database(s). As part of managing the database(s), the stock inventory module 310b

may identify components that are damaged or otherwise unavailable for use in structural designs. For example, the stock inventory module 310b may receive user input (e.g., via user input device(s) 302) specifying damaged components, or may identify damaged components based on data in the stock file(s).

5 One or more implementations of the association module 310c work in connection with the design module 310e to associate stock components with components of a structural design. For example, the association module 310c may automatically match components of a structural design imported using import module 310a to stock components. In another example, the association module 310c may
10 associate newly placed components with stock components.

 One or more implementations of the import module 310d operate to import data into the design software application 310. For example, the import module 310d may import structural designs and stock files. In some situations, the import module 310d may import data from the storage device(s) 306. The import module 310d may
15 also import data from remote computer systems, such as over a network. One or more implementations of the user interface module 310e generate any user interfaces of the design software application 310. For example, the user interface module 310e may generate the example user interfaces 100, 200 depicted in Figures 1 and 2.

 In addition to the foregoing, the design software system can identify each
20 component's location, type, size, relationships to other components, etc. Upon identification of the components, the design software system can create intelligent, automatically resolvable objects for each component that includes one or more attributes and product parameters.

 The design software system can create and compile objects in virtually any
25 object-oriented programming language, including, for example, an object-oriented programming language such as the JAVA programming language. No matter how created, the design software system can store the objects within a single data structure, and resolve each object (where applicable) in the design to conform to the given design intent. The design software system can thus appropriately propagate
30 user input at any level of this dataset to all objects in the data structure consistent with the design intent, thereby resolving the complete design through the resolving of each

object individually. These programming objects can include new components and stock components whether they are individual parts or assemblies.

Each object in the database can include one or more attributes and product parameters, or reference to one or more databases comprising information of the same. Attributes can include product features and characteristics, such as, for
5 example, size, shape, color, type, texture, price, material composition, etc. Product parameters can include rules and behaviors associated with each component. For example, the rules and behaviors can include the type and number of connectors and bracketry need to support each component or assembly, how the components and
10 assemblies interact with other design elements, applicable codes, manufacturing guidelines, and physical limitations.

Figures 1-3 provide a number of user interfaces, schematics, and mechanisms for incorporating stock components into structural designs. In addition to the foregoing, implementations of the present invention can also be described in terms of
15 flowcharts comprising one or more acts in a method for accomplishing a particular result. For example, Figure 4 illustrates a method 400 of visually modeling a new structural design, in real-time, at a structural design software application. Similarly, Figure 5 illustrates a method 500 of modifying a structural design, in real-time, at a structural design software application. The acts of Figures 4 and 5 are discussed more
20 fully below with respect to the user interfaces and schematics of Figures 1-3. Of course, as a preliminary manner, one of ordinary skill in the art will recognize that the methods explained in detail herein can be modified. For example various acts of each method described can be omitted or expanded, and the order of the various acts of each method described can be altered as desired.

For example, Figure 4 shows that the method 400 of visually modeling a new
25 structural design, in real-time, at a structural design software application can comprise an act 402 of importing a stock file. Act 402 can include importing a stock file, the stock file including a plurality of existing components that are used in an existing structural design, the plurality of existing components being available for use in the
30 new structural design. For example, the import module 310d can import a stock file. The stock file may comprise an existing structural design, a data file generated from an existing structural design, a spreadsheet, etc. A designer may trigger use of the

import module 310d using a user interface control, such as the button 204 (manage stock) of the user interface 200.

In addition, Figure 4 shows that the method 400 in accordance with an implementation of the present invention can comprise an act 404 of identifying
5 existing components that are in the stock file. Act 404 can include identifying each of the plurality of existing components that are in the stock file. For example, the stock inventory module 310b can analyze the stock file imported during act 402 and identify stock components (and possibly assemblies) that are included in the stock file.

10 Figure 4 also shows that the method 400 in accordance with an implementation of the present invention can comprise an act 406 of receiving user input specifying a design element. Act 406 can include receiving user input that specifies a new design element of the new structural design. For example, user interface module 310e can receive user input specifying a new component via the user
15 interface 100 of Figure 1. For example, a designer may use the inventory browser 102 to select a sub-category (e.g., doors) of the “products” category, and then use the component selector 104 to select the new component (e.g., a new door). The designer may place the new component into a structural design within the design area 106. The design module 310a can receive notice of the placement of the new component.

20 Furthermore, Figure 4 shows that the method 400 in accordance with an implementation of the present invention can comprise an act 408 of identifying that existing components correspond with the design element. Act 408 can include automatically identifying that one or more of the plurality of existing components correspond with one or more components of the new design element. For example,
25 the design module 310a may trigger the association module 310c. The association module 310c can, in turn, consult a stock inventory database directly or work with the stock inventory module 310b to determine if the new component matches any stock components from the stock file imported during act 402.

30 Still further, Figure 4 shows that the method 400 in accordance with an implementation of the present invention can comprise an act 410 of associating the existing components with the design element. Act 410 can include associating the automatically identified one or more of the plurality of existing components with the

new structural design. For example, the design module 310a can incorporate the stock component into the structural design instead of the requested new component. In addition, the stock inventory module 310b can update the stock inventory database to reflect use of the stock component in the structural design. Updating the stock
5 inventory database may update one or more user interfaces. For example, when displaying the stock component in the component selector 104 of user interface 100, the user interface module 310e may indicate that at least one of the stock components is in use. In addition, the user interface 200 of Figure 2 may also reflect use of the stock component in the structural design.

10 Additionally, Figure 4 shows that the method 400 in accordance with an implementation of the present invention can comprise an act 412 of visually distinguishing between existing components and new components. Act 412 can include visually distinguishing between any existing components that are associated with the new structural design and any new components that are associated with the
15 new structural design. For example, when displaying design area 106, user interface module 310e can use different colors or different shading to indicate which components of the displayed design are from stock, and which components are new. In addition, the user interface 200 of Figure 2 may highlight rows corresponding to stock components using one color (or one set of colors), and rows corresponding to
20 new components in a different color (or different set of colors).

In addition to the method 400 of Figure 4, Figure 5 shows that the method 500 of modifying a structural design in, real-time, at a structural design software application can comprise an act 502 of receiving a structural design. Act 502 can include receiving a first structural design. For example, the import module 310d can
25 import a structural design file.

Figure 5 also shows that the method 500 in accordance with an implementation of the present invention can comprise an act 504 of receiving a stock file. Act 504 can include receiving a stock file, the stock file including a plurality of existing components that are used in an existing second structural design, the plurality
30 of existing components being available for use in the first structural design. For example, the import module 310d can import a stock file. The stock file may comprise the existing structural design itself, a data file generated from the existing

structural design, a spreadsheet, etc. A designer may trigger use of the import module 310d with a user interface control, such as the button 204 (manage stock) of user interface 200.

Figure 5 also shows that the method 500 in accordance with an implementation of the present invention can comprise an act 506 of identifying existing components that are in the stock file. Act 506 can include identifying each of the plurality of existing components that are in the stock file. For example, the stock inventory module 310b can analyze the stock file imported during act 504 and identify stock components (and possibly assemblies) that are included in the stock file.

Figure 5 also shows that the method 500 in accordance with an implementation of the present invention can comprise an act 508 of identifying existing components that are usable with the structural design. Act 508 can include identifying one or more of the plurality of existing components that are usable the first structural design. For example, the association module 310c can compare the structural design imported in act 502 with stock components included in the stock file imported in act 504. When doing so, the association module 310c can identify matches between components in the imported structural design and the stock components. Matches may comprise exact matches that match in both type and style, or similar matches that match in type but not style.

Figure 5 also shows that the method 500 in accordance with an implementation of the present invention can comprise an act 510 of associating the identified existing components with the structural design. Act 510 can include associating the identified one or more of the plurality of existing components with the first structural design. For example, the design module 310a can associate any matching stock components identified in act 508 with the imported structural design. As such, with very little effort, a designer can incorporate existing stock into an already-created design.

Accordingly, Figures 1-5 provide a number of components, user interfaces, schematics, and mechanisms for easy and efficient structural design and/or reconfiguration while reusing existing stock components. One or more implementations allow a designer to import a stock file containing an inventory of

stock components that are usable as part of a structural design. The implementations described herein incorporate the stock components in a structural design while tracking, in real-time, the availability of stock components. One or more implementations automatically correlate stock components with components already
5 inserted into a structural design. One or more implementations automatically correlate stock components with new components as a designer inserts them into the structural design. One or more implementations visually distinguish between new components in the structural design and stock components in the structural design.

Implementations of the present invention may comprise or utilize a special
10 purpose or general-purpose computer including computer hardware, such as, for example, one or more processors and system memory, as discussed in greater detail below. Implementations within the scope of the present invention also include physical and other computer-readable media for carrying or storing computer-executable instructions and/or data structures. Such computer-readable media can be
15 any available media that can be accessed by a general purpose or special purpose computer system. Computer-readable media that store computer-executable instructions are computer storage media (devices). Computer-readable media that carry computer-executable instructions are transmission media. Thus, by way of example, and not limitation, implementations of the invention can comprise at least
20 two distinctly different kinds of computer-readable media: computer storage media (devices) and transmission media.

Computer storage media (devices) includes RAM, ROM, EEPROM, CD-ROM, solid state drives (“SSDs”) (e.g., based on RAM), Flash memory, phase-change memory (“PCM”), other types of memory, other optical disk storage,
25 magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer.

A “network” is defined as one or more data links that enable the transport of
30 electronic data between computer systems and/or modules and/or other electronic devices. When information is transferred or provided over a network or another communications connection (either hardwired, wireless, or a combination of

hardwired or wireless) to a computer, the computer properly views the connection as a transmission medium. Transmissions media can include a network and/or data links which can be used to carry desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer. Combinations of the above should also be included within the scope of computer-readable media.

Further, upon reaching various computer system components, program code means in the form of computer-executable instructions or data structures can be transferred automatically from transmission media to computer storage media (devices) (or vice versa). For example, computer-executable instructions or data structures received over a network or data link can be buffered in RAM within a network interface module (e.g., a “NIC”), and then eventually transferred to computer system RAM and/or to less volatile computer storage media (devices) at a computer system. Thus, it should be understood that computer storage media (devices) can be included in computer system components that also (or even primarily) utilize transmission media.

Computer-executable instructions comprise, for example, instructions and data which, when executed at a processor, cause a general purpose computer, special purpose computer, or special purpose processing device to perform a certain function or group of functions. The computer executable instructions may be, for example, binaries, intermediate format instructions such as assembly language, or even source code. Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the described features or acts described above. Rather, the described features and acts are disclosed as example forms of implementing the claims.

Those skilled in the art will appreciate that the invention may be practiced in network computing environments with many types of computer system configurations, including, personal computers, desktop computers, laptop computers, message processors, hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, mobile telephones, PDAs, tablets, pagers, routers, switches,

and the like. The invention may also be practiced in distributed system environments where local and remote computer systems, which are linked (either by hardwired data links, wireless data links, or by a combination of hardwired and wireless data links) through a network, both perform tasks. In a distributed system environment, program
5 modules may be located in both local and remote memory storage devices.

The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the
10 foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

CLAIMS

I claim:

1. At a computer system that includes one or more processors and system memory, a method for visually modeling a new structural design, in real-time, at a structural design software application, the method comprising acts of:
 - importing a stock file, the stock file including a plurality of existing components that are used in an existing structural design, the plurality of existing components being available for use in the new structural design;
 - identifying each of the plurality of existing components that are in the stock file;
 - receiving user input that specifies a new design element of the new structural design;
 - automatically identifying that one or more of the plurality of existing components corresponds with one or more components of the new design element;
 - associating the automatically identified one or more of the plurality of existing components with the new structural design; and
 - visually distinguishing between any existing components that are associated with the new structural design and any new components that are associated with the new structural design.
2. The method as recited in claim 1, further comprising an act of receiving user input that identifies one or more of the plurality of components as being damaged, and wherein the act of automatically identifying that one or more of the plurality of existing components correspond with one or more components of the new design element comprises an act of excluding any of the plurality of components that are identified as being damaged.
3. The method as recited in claim 1, further comprising acts of:
 - visually presenting an inventory of the plurality of existing components that are in the stock file; and
 - updating the inventory as existing components from the plurality of existing components are associated with the new structural design.

4. The method as recited in claim 1, wherein the act of visually distinguishing between any existing components that are associated with the new structural design and any new components that are associated with the new structural design comprises an act of using different colors to distinguish between existing
5 components and new components.

5. The method as recited in claim 1, wherein the act of associating the automatically identified one or more of the plurality of existing components with the new structural design comprises an act of automatically incorporating the identified one or more of the plurality of existing components into the new structural design by
10 replacing the one or more components of the new design element.

6. The method as recited in claim 1, wherein the act of automatically identifying that one or more of the plurality of existing components correspond with one or more components of the new design element comprises an act of automatically identifying an exact match between one or more existing components and the one or
15 more components of the new design element, an exact match comprising an exact match between component type and visual style.

7. The method as recited in claim 1, wherein the act of automatically identifying that one or more of the plurality of existing components correspond with one or more components of the new design element comprises an act of automatically
20 identifying a similar match between one or more existing components and the one or more components of the new design element, a similar match comprising a match between components having similar function but different styles.

8. The method as recited in claim 1, wherein the act of associating the automatically identified one or more of the plurality of existing components with the
25 new structural design comprises an act of automatically adjusting one or more new components to match at least one existing component, including automatically adjusting or more of a size of the one or more new components or a style of the one or more new components.

9. The method as recited in claim 1, further comprising acts of:
visually presenting a percentage of existing components used in the
new structural design; and
dynamically updating the percentage as the new structural design is
5 modified.

10. The method as recited in claim 1, wherein the design element of the
new structural design comprises an assembly that includes a plurality of components,
and wherein the act of automatically identifying that one or more of the plurality of
existing components correspond with one or more components of the new design
10 element comprises an act of identifying a plurality of existing components that are
usable for the plurality of components of the assembly.

11. At a computer system including one or more processors and system
memory, a method for modifying a structural design, in real-time, at a structural
design software application, the method comprising acts of:

15 receiving a first structural design;
receiving a stock file, the stock file including a plurality of existing
components that are used in an existing second structural design, the plurality
of existing components being available for use in the first structural design;
identifying each of the plurality of existing components that are in the
20 stock file;
identifying one or more of the plurality of existing components that are
usable the first structural design; and
associating the identified one or more of the plurality of existing
components with the first structural design.

25 12. The method as recited in claim 11, wherein the act of identifying one
or more of the plurality of existing components that are usable the first structural
design comprises identifying existing components that are exact matches, an exact
match comprising an exact match between component type and visual style.

30 13. The method as recited in claim 11, wherein the act of identifying one
or more of the plurality of existing components that are usable the first structural
design comprises identifying existing components that are close matches, a close
match comprising a match between components having similar function.

14. The method as recited in claim 13, wherein a close match also comprises a match between components having different aesthetic styles.

15. The method as recited in claim 11, wherein the acts of identifying one or more of the plurality of existing components that are usable the first structural design and associating the identified one or more of the plurality of existing components with the first structural design are performed automatically by the computer system.

16. The method as recited in claim 11, further comprising acts of:
visually presenting an inventory of the plurality of existing components that are in the stock file; and
updating the inventory as existing components from the plurality of existing components are placed in the first structural design.

17. The method as recited in claim 11, further comprising an act of visually distinguishing between any existing components that are associated with the first structural design and any new components that are associated with the first structural design.

18. The method as recited in claim 17, wherein the act of visually distinguishing between any existing components that are associated with the first structural design and any new components that are associated with the first structural design comprises an act of providing using a first color for existing components and a second color for new components.

19. The method as recited in claim 11, further comprising an act of automatically modifying one or more of a size or a type of at least one component in the first structural design to accommodate an existing component.

20. A computer program product for use at a computer system, the computer program product for implementing a method for visually creating a new structural design model at a structural design software application, the computer program product comprising one or more computer storage media having stored thereon computer-executable instructions that, when executed at a processor, cause the computer system to perform the method, including the following:

- importing a stock file, the stock file identifying a plurality of existing components that are used in an existing structural design, the plurality of existing components being available for use in the new structural design;
- 10 identifying each of the plurality of existing components that are in the stock file;
- receiving user input that specifies a new design element of the new structural design;
- 15 automatically identifying that one or more of the plurality of existing components correspond with one or more components of the new design element;
- associating the automatically identified one or more of the plurality of existing components with the new structural design; and
- 20 visually distinguishing between any existing components that are associated with the new structural design and any new components that are associated with the new structural design.

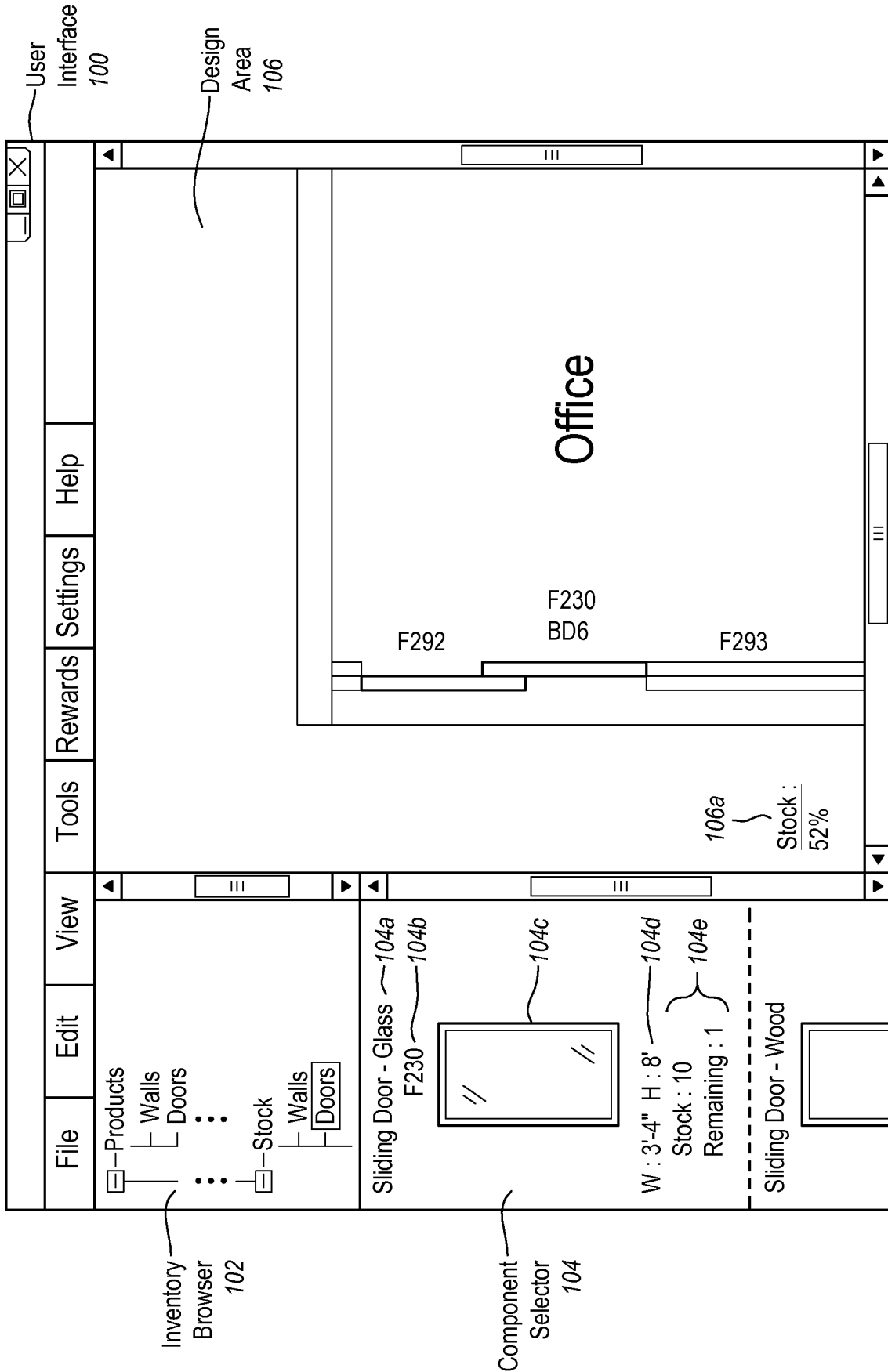


Fig. 1

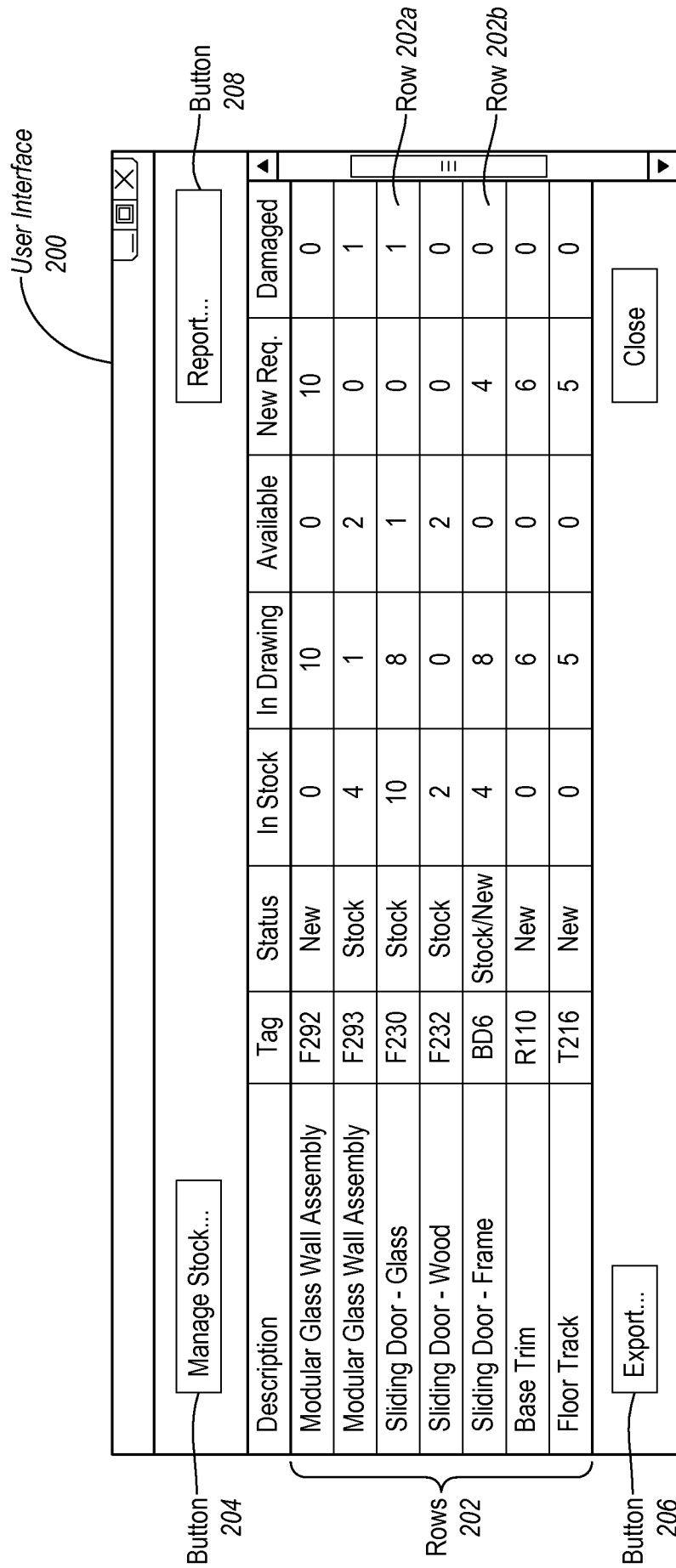


Fig. 2

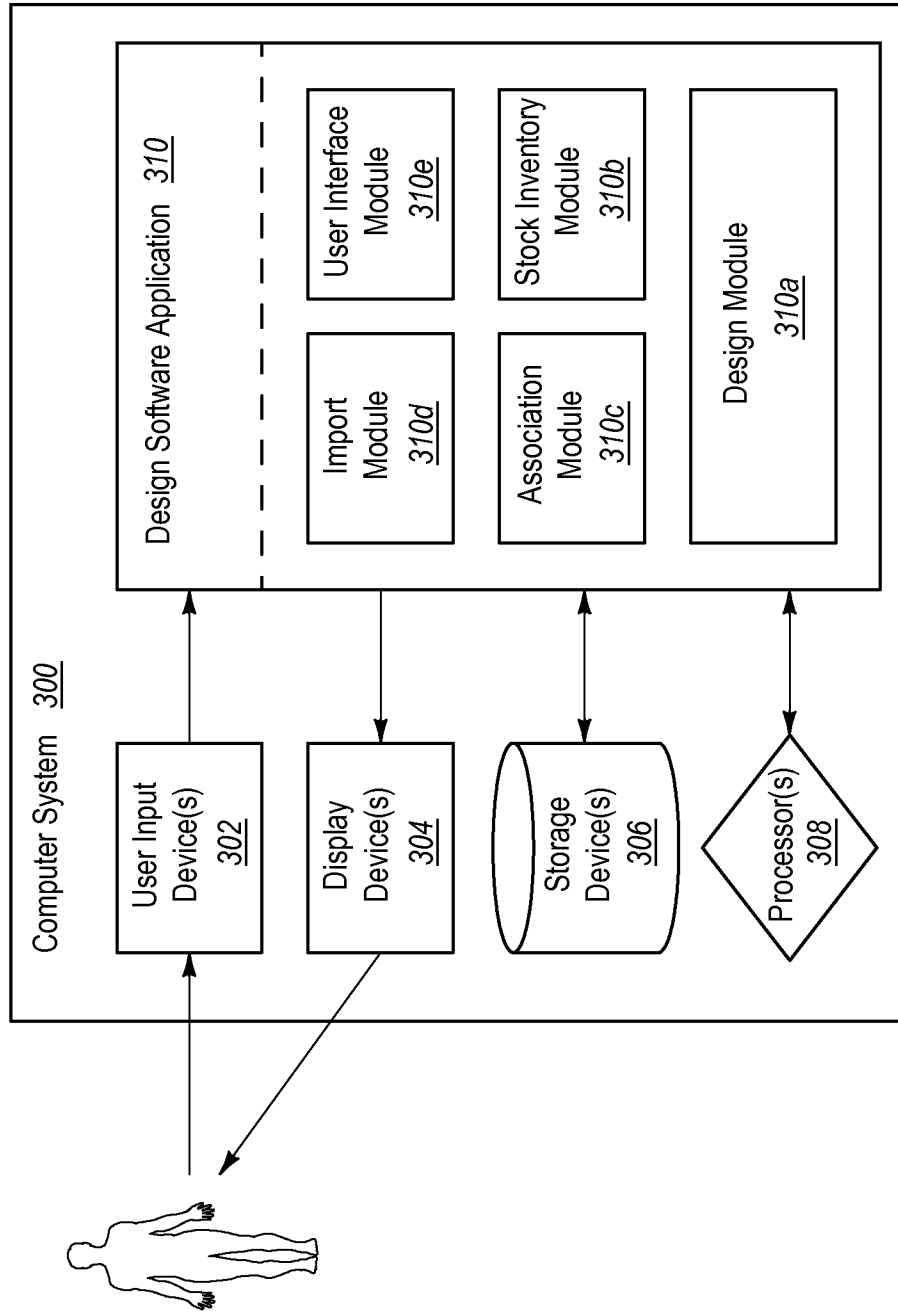
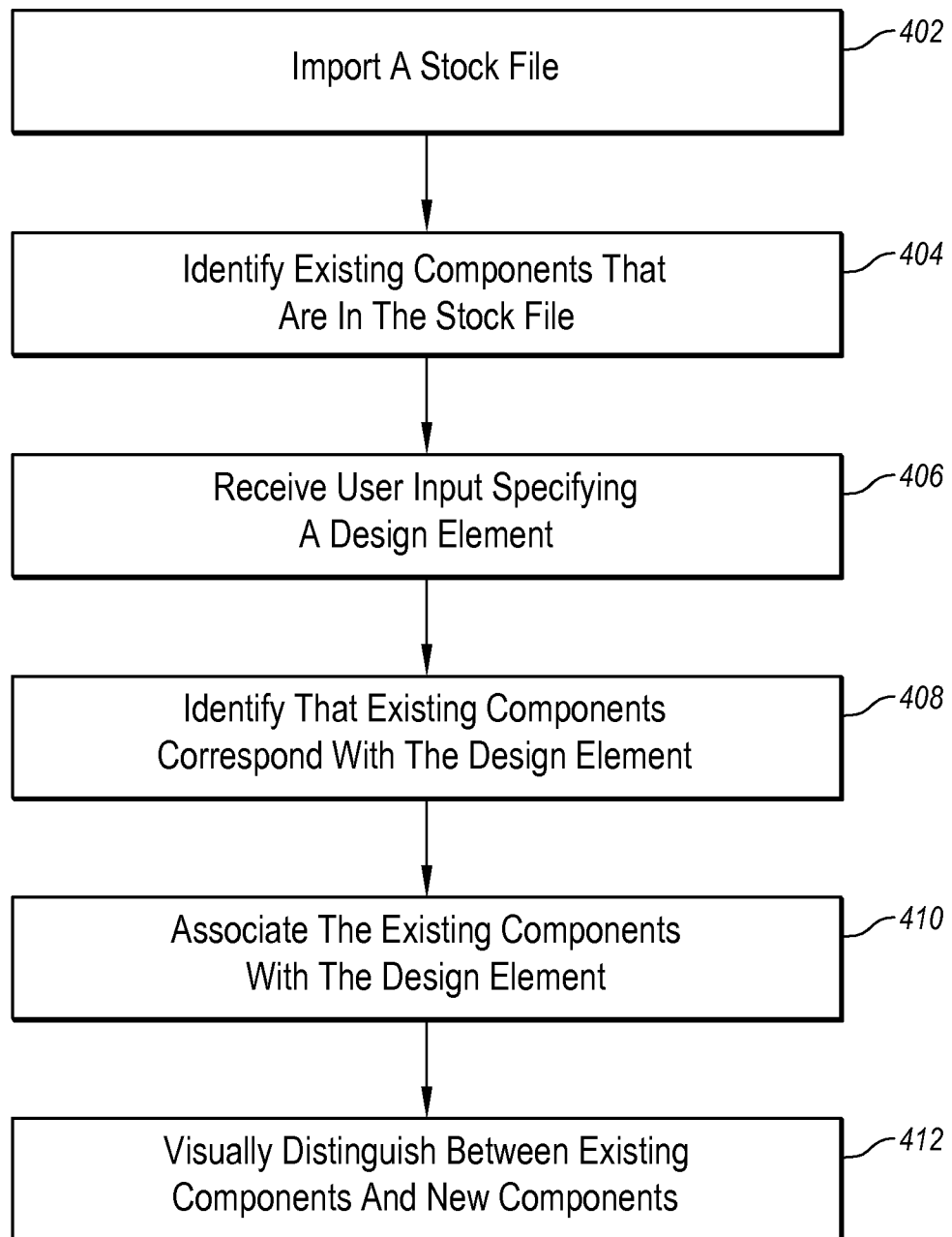
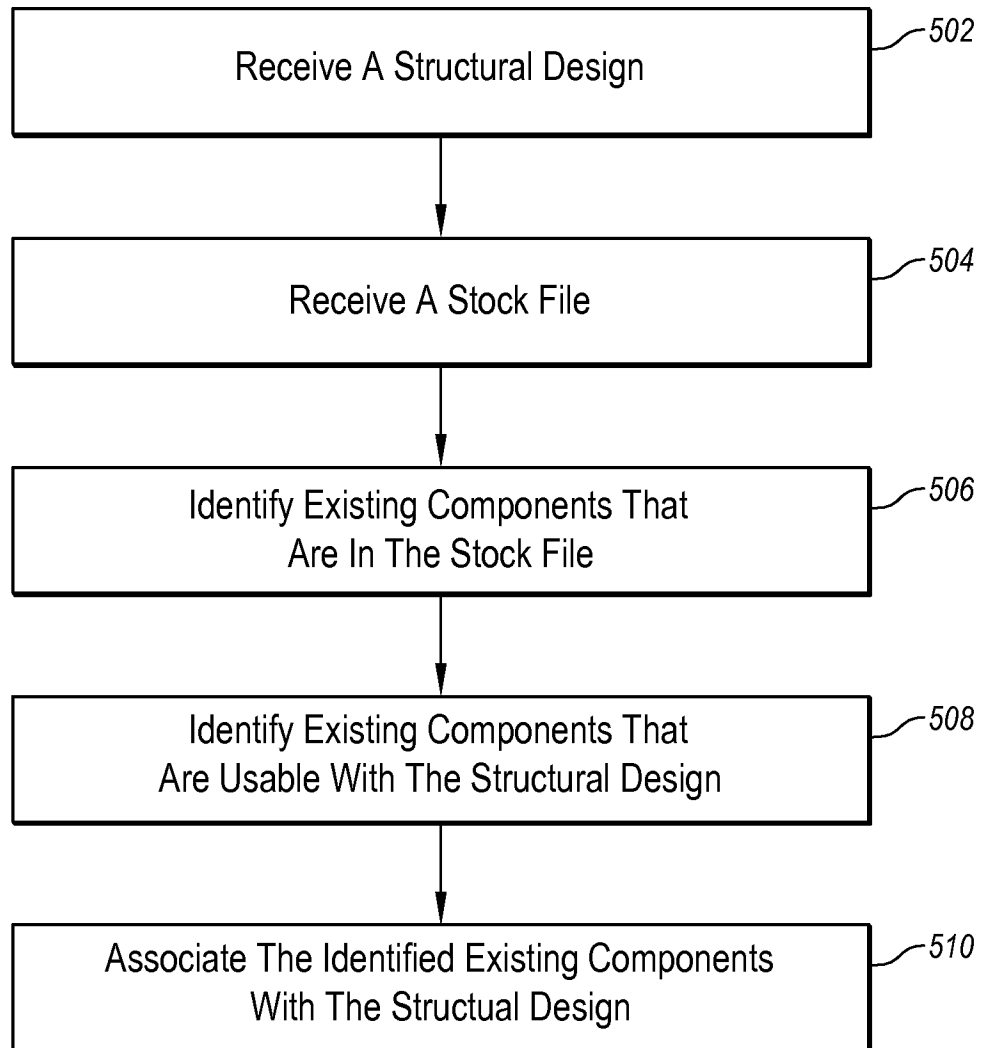


Fig. 3

4 / 5

**Fig. 4**

5 / 5

**Fig. 5**