CLAIMS

1. In a computerized environment, a computerized method for incorporating third party features into an architectural design, the method comprising:

generating an architectural design comprising one or more sub-components, wherein each of the one or more sub-components is associated with a different independently executable software object that is unique to each sub-component selected from the one or more sub-components and each different independently executable software object comprises independent variables and functions that are specific to different architectural elements that are represented by each sub-component selected from the one or more sub-components;

identifying a hardware interface of a third party feature with at least one sub-component of the architectural design, wherein the hardware interface is configured to receive the third party feature;

receiving an indication of specifications of the third party feature; and

automatically incorporating the specifications of the third party feature into the hardware interface, wherein:

the hardware interface is adaptable to receive a plurality of different third party features,

a particular independently executable software object that is unique to the at least one sub-component of the architectural design provides direction about the correct placement of the third party feature with respect to the hardware interface, wherein the particular independently executable software object inherited its characteristics from a parent independently executable software object from which the particular independently executable software object was split;

the specifications of the third party feature are the particular specifications of a millwork facility that will manufacture the hardware interface, and

automatically incorporating the specifications of the third party feature comprises automatically configuring a numerical control file to incorporate the specifications of the third party feature into the hardware interface.

2. The method as recited in claim 1, wherein identifying the interface comprises:

identifying a first sub-component and a second sub-component;

determining that a particular area of the first sub-component will be in physical communication with the second sub-component; and

associating the interface with the particular area.

3. The method as recited in claim 2, wherein the third party feature comprises a particular carpentry joint.

4. The method as in claim 3, wherein the particular carpentry joint comprises a joint selected from the group consisting of a bridle joint, a scarf joint, a butt joint, a miter joint, a lap joint, a box joint, a dovetail joint, a dado joint, a groove joint, a tongue and groove joint, a mortise and tenon joint, a bridsmouth joint, a finger joint, a halved joint, and a splice joint.

5. The method as recited in claim 1, wherein identifying the interface comprises:

identifying a first sub-component;

determining that the first sub-component will be in physical communication with a hardware component; and

associating the interface with the first sub-component.

6. The method as recited in claim 5, wherein the third party feature comprises a particular hardware component.

7. The method as recited in claim 1, further comprising:

receiving a first configuration file, wherein the first configuration file comprises a first listing of third-party features that are associated with a particular manufacturer; and

automatically determining that the interface is associated with a specific third-party feature listed in the first configuration file.

8. The method as recited in claim 7, further comprising:

receiving a second configuration file, wherein the second configuration file comprises a different listing of third party features that are associated with another manufacturer; and

automatically determining that the interface is associated with a specific third-party feature listed in the second configuration file.

9. The method as recited in claim 1, further comprising:

automatically resolving anomalies created by the incorporation of the third party feature.

10. The method as recited in claim 9, wherein automatically resolving anomalies comprises:

adjusting a size of the at least one sub-component of the architectural design to accommodate the third party feature.

11. A computer system for incorporating third party components into an architectural design, comprising:

one or more processors; and

one or more computer-readable media having stored thereon executable instructions that when executed by the one or more processors configure the computer system to perform at least the following:

generate within a computer aided drafting program an architectural design for an architectural element, wherein the architectural design comprises one or more sub-components, each sub-components within the one or more sub-components is associated with a different independently executable software objects, each independently executable software object comprising independent variables and functions that are specific to different architectural elements that are represented by each sub-component;

identify at least one interface for manufacturer specific components within the architectural element, wherein the at least one interface is associated with a particular independently executable software object that comprises independent variables and functions that are specific to a particular architectural element that comprises the at least one interface, wherein the particular independently executable software object inherited its characteristics from a parent independently executable software object from which the particular independently executable software object was split;

receive a configuration list, wherein the configuration list comprises manufacturer specific components and manufacturer specific specifications for a particular manufacturer;

automatically identify a manufacturer specific component and manufacturer specific specifications from the configuration list that correspond with the at least one interface; and

automatically incorporate the manufacturer specific component into the architectural design, such that the design represents a mixture of user design format and formatting from the incorporated manufacturer-specific component and specifications, wherein the independently executable software object provides direction about the correct placement of the manufacturer specific component.

12. The system as recited in claim 11, wherein automatically incorporating the manufacturer specific components into the architectural design, comprises:

determining that the manufacturer specific component requires a third party feature to be incorporated within the architectural design; and

automatically adjusting the architectural design to incorporate the third party feature.

13. The system as recited in claim 12, wherein the executable instructions include instructions that are executable to configure the computer system to:

automatically incorporate the third party feature into a computer numerical control file.

14. The system as recited in claim 13, wherein the third party feature comprises a specific carpentry joint.

15. The system as recited in claim 13, wherein the third party feature requires a specific millwork operation.

16. The system as recited in claim 11, wherein the executable instructions include instructions that are executable to configure the computer system to:

automatically resolve an inconsistency that is generated by the incorporation of the manufacturer specific component into the architectural design.

17. The system as recited in claim 16, wherein automatically resolving an inconsistency comprises:

identifying that the manufacturer specific component is not properly sized to fit within the at least one interface for the manufacturer specific component; and

automatically adjusting the architectural design to properly incorporate the manufacturer specific component.

18. The system as recited in claim 16, wherein automatically resolving an inconsistency comprises:

identifying that the manufacturer specific component comprises a specific type of carpentry joint;

determining that the incorporation of the specific type of carpentry joint creates an anomaly within the architectural design; and

automatically adjusting the architectural design to resolve the anomaly.

19. The system as recited in claim 11, wherein the same architectural design is used independently by a first manufacturer and a second manufacturer, such that:

the architectural design relies upon a first configuration list to automatically incorporate a first set of manufacturer specific components from the first manufacturer into the architectural design; and

the architectural design relies upon a second configuration list to automatically incorporate a second set of manufacturer specific components from the second manufacturer into the architectural design.

20. A computer-based system comprising one or more processors configured to execute computer-executable instructions for a method for incorporating third party features into an architectural design, the method comprising:

generating an architectural design comprising one or more sub-components, wherein each of the one or more sub-components is associated with a different independently executable software object that is unique to each sub-component selected from the one or more sub-components and each different independently executable software object comprises independent variables and functions that are specific to different architectural elements that are represented by each sub-component selected from the one or more sub-components;

identifying a hardware interface of a third party feature with at least one sub-component of the architectural design, wherein the hardware interface is configured to receive the third party feature;

receiving specifications of the third party feature;

automatically incorporating the third party feature into the interface, wherein:

a particular independently executable software object that is unique to the at least one sub-component of the architectural design provides direction about the correct placement of the third party feature with respect to the hardware interface, wherein the particular independently executable software object inherited its characteristics from a parent independently executable software object from which the particular independently executable software object was split; and

automatically resolving anomalies that the incorporation of the third party feature creates.