CLAIMS

1. A system for visual software development, comprising an application interface layer and a data processing layer; wherein

the application interface layer is configured to display N module displaying controls, wherein the N module displaying controls correspond to N modules respectively, module data of each of the N modules is stored in the data processing layer, and the module data comprises an execution logic of a corresponding module, wherein the application interface layer is further configured to display a module adding control, which is configured to import module data of a third-party module to the data processing layer, wherein the third-party module is a module independently designed and developed by a third party, and the N modules comprise the third-party module, and N is a positive integer greater than or equal to 1;

wherein the application interface layer is further configured to display a flow chart in response to a first operation of a user on one or more module displaying controls in the N module displaying controls, the flow chart indicating an execution order of modules respectively corresponding to the one or more module displaying controls; and

wherein the data processing layer is configured to call and run the module data of the modules respectively corresponding to the one or more module displaying controls in response to the flow chart displayed by the application interface layer, to obtain a data processing result of the flow chart, and return the data processing result to the application interface layer;

wherein the application interface layer is configured to acquire the module data of the third-party module imported by the user through the module adding control in response to a second operation on the module adding control, and send the imported module data of the third-party module to the data processing layer; and

the data processing layer is configured to store the imported module data of the third-party module when the imported module data of the third-party module is

verified to meet a reference rule, and return a module loading success message to the application interface layer, wherein the module loading success message indicates that the third-party module is successfully loaded.

2. The system according to claim 1, wherein the reference rule comprises a design mode of an input/output interface of the third-party module and/or a package mode of the module data of the third-party module;

wherein the design mode of the input/output interface of the third-party module comprises a format of input data and/or output data of the third-party module, wherein the package mode of the module data of the third-party module indicates a storage format of the module data of the third-party module in the data processing layer, and the storage format comprises any one of a dynamic link library, static link library, exe file.

3. The system according to claim 2, wherein the application interface layer is further configured to display a module configuration file indicating the reference rule in response to a third operation on the module adding control.

4. The system according to claim 1, wherein the application interface layer is further configured to display a module configuration interface of a first module corresponding to a first module displaying control in the N module displaying controls in response to a fourth operation of the user on the first module displaying control, wherein the module configuration interface comprises a plurality of module configuration controls, and any one of the plurality of the module configuration controls is configured to instruct the user to edit a parameter(s) in execution of the first module;

the application interface layer is further configured to: in response to a fifth operation on the any one of the module configuration controls, acquire a parameter(s) edited by the user through the any one of the module configuration controls and send the edited parameter(s) to the data processing layer;

the data processing layer is further configured to update the module data of the first module based on the edited parameter(s); and

wherein when the flow chart comprises the first module, the data processing layer is further configured to update the data processing result of the flow chart based on the updated module data of the first module, and return the updated data processing result to the application interface layer.

5. The system according to claim 4, wherein when the first module is a third-party module, the module data of the first module imported by the user through the module adding control further comprises a module configuration interface file, which indicates a program code of the module configuration interface of the first module; and

the module configuration interface of the first module displayed by the application interface layer is displayed according to the module configuration interface file in the module data of the first module.

6. The system according to claim 1, wherein the data processing layer is further configured to generate a first code package based on the flow chart, which indicates an execution logic of each of modules comprised in the flow chart and an execution order of the modules comprised in the flow chart; and

the data processing layer is further configured to control the application interface layer to display a first code control corresponding to the first code package, wherein the first code control is configured to instruct the user to download the first code package, and the first code package is allowed to be imported into a third-party system and enables the flow chart to be displayed after being run on the third-party system.

7. The system according to claim 1, wherein the application interface layer is further configured to display a running interface displaying control, wherein the running interface displaying control indicates the displaying of a running interface for the flow chart, the running interface is configured to display running information of the flow chart, and the running information of the flow chart indicates intermediate data in running of the flow chart; and

the application interface layer is further configured to display the running

interface in response to a sixth operation on the running interface displaying control, to enable the user to view the intermediate data in running of the flow chart based on the running interface.

8. The system according to claim 7, wherein the application interface layer is further configured to display a running interface configuration control, wherein the running interface configuration control indicates the displaying of a running interface configuration interface, which is configured to configure contents displayed on the running interface;

the application interface layer is further configured to display the running interface configuration interface in response to a seventh operation on the running interface configuration control, wherein the running interface configuration interface comprises a plurality of running interface configuration sub-controls, and any one of the plurality of running interface configuration sub-controls corresponds to an open option;

the application interface layer is further configured to display a content source option in response to an eighth operation on the open option of a first running interface configuration sub-control of the plurality of running interface configuration sub-controls, and to bind the first running interface configuration sub-control with one module in the flow chart in response to a ninth operation on the content source option; and

wherein, the running information displayed on the running interface comprises data of the module bound to the first running interface configuration sub-control.

9. The system according to claim 7, wherein the data processing layer is further configured to export a program code of the running interface as a second code package; and

the data processing layer is further configured to control the application interface layer to display a second code control corresponding to the second code package, wherein the second code control is configured to instruct the user to download the second code package, and the second code package is allowed to be imported into a

third-party system and enables the running interface to be displayed after being run on the third-party system.

10. The system according to any one of claims 1 to 9, wherein the data processing layer comprises a data scheduling layer and a functional module layer;

module data of the N modules is stored in the functional module layer; and

the data scheduling layer is configured to sequentially call and run the module data of the modules respectively corresponding to the one or more module displaying controls from the functional module layer according to the execution order of the modules respectively corresponding to the one or more module displaying controls in the flow chart, to obtain the data processing result.

11. The system according to claim 1, wherein the data processing layer is configured to store module data of M modules, wherein M is a positive integer greater than or equal to N; and

the application interface layer is further configured to display a module registration control, and the application interface layer is configured to display the N module displaying controls in response to the N modules selected by the user from the M modules through the module registration control.

12. The system according to claim 11, wherein the N modules are modules verified by the data processing layer to be valid, and the modules verified to be valid refer to corresponding modules that the user has permission to use.

13. The system according to claim 1, wherein the application interface layer is further configured to display a calling control for any one of the module displaying controls, wherein the calling control indicates that the user is allowed to call a module corresponding to the any one of the module displaying controls in a third-party system;

the application interface layer is further configured to display a port link of module data of the module corresponding to the any one of the module displaying controls in response to a tenth operation on the calling control;

the application interface layer is further configured to acquire the module data of the module corresponding to the any one of the module displaying controls from the data processing layer based on the port link when a data calling request for the port link sent by the third-party system is detected, and send the acquired module data to the third-party system.

14. A method for visual software development, which is applied to a system for visual software development comprising an application interface layer and a data processing layer, the method comprising: displaying, by the application interface layer, N module displaying controls, wherein the N module displaying controls correspond to N modules respectively, module data of each of the N modules is stored in the data processing layer, and the module data comprises an execution logic of a corresponding module; further displaying, by the application interface layer, a module adding control, which is configured to import module data of a third-party module to the data processing layer, wherein the third-party module is a module independently designed and developed by a third party, wherein the N modules comprise the third-party module, and N is a positive integer greater than or equal to 1; displaying, by the application interface layer, a flow chart in response to a first operation of a user on one or more module displaying controls in the N module displaying controls, wherein the flow chart indicates an execution order of modules respectively corresponding to the one or more module displaying controls; calling and running, by the data processing layer, the module data of the modules respectively corresponding to the one or more module displaying controls in response to the flow chart displayed by the application interface layer, to obtain a data processing result of the flow chart, and returning, by the data processing layer, the data processing result to the application interface layer;

wherein the method further comprises: acquiring, by the application interface layer, the module data of the third-party module imported by the user through the module adding control in response to a second operation on the module adding control, and sending, by the application interface layer, the imported module data of the third-party module to the data processing layer; storing, by the data processing layer, the imported module data of the third-party module when verifying the imported

module data of the third-party module to meet a reference rule, and returning, by the data processing layer, a module loading success message to the application interface layer, wherein the module loading success message indicates that the third-party module is successfully loaded.

15. The method according to claim 14, wherein the reference rule comprises a design mode of an input/output interface of the third-party module and/or a package mode of the module data of the third-party module; wherein the design mode of the input/output interface of the third-party module comprises a format of input data and/or output data of the third-party module, wherein the package mode of the module data of the third-party module indicates a storage format of the module data of the third-party module in the data processing layer, and the storage format comprises any one of a dynamic link library, static link library and exe file.

16. The method according to claim 14 or 15, wherein the method further comprises: displaying, by the application interface layer, a module configuration file indicating the reference rule in response to a third operation on the module adding control.

17. The method according to claim 14, wherein the method further comprises: displaying, by the application interface layer, a module configuration interface of a first module corresponding to a first module displaying control in the N module displaying controls in response to a fourth operation of the user on the first module displaying control, wherein the module configuration interface comprises a plurality of module configuration controls, and any one of the plurality of module configuration controls is configured to instruct the user to edit a parameter(s) in execution of the first module; acquiring, by the application interface layer, a parameter(s) edited by the user through the any one of the module configuration controls in response to a fifth operation on the any one of the module configuration controls, and sending, by the application interface layer, the edited parameter(s) to the data processing layer; and updating, by the data processing layer, the module data of the first module based on the edited parameter(s); wherein when the flow chart comprises the first module, the data processing layer updates the data processing

result of the flow chart based on the updated module data of the first module, and returns the updated data processing result to the application interface layer.

18. The method according to claim 17, wherein when the first module is a third-party module, the module data of a first module imported by the user through the module adding control further comprises a module configuration interface file, which indicates a program code of a module configuration interface of the first module; and wherein the module configuration interface of the first module displayed by the application interface layer is displayed according to the module configuration interface file in the module data of the first module.

19. The method according to claim 14, wherein the method further comprises: generating, by the data processing layer, a first code package based on the flow chart, which indicates an execution logic of each of modules comprised in the flow chart and an execution order of the modules comprised in the flow chart; and controlling, by the data processing layer, the application interface layer to display a first code control corresponding to the first code package, wherein the first code control is configured to instruct the user to download the first code package, and the first code package is allowed to be imported into a third-party system and enables the flow chart to be displayed after being run on the third-party system.

20. The method according to claim 14, wherein the method further comprises: displaying, by the application interface layer, a running interface displaying control, wherein the running interface displaying control indicates the displaying of a running interface for the flow chart, and the running interface is configured to display running information of the flow chart, and the running information of the flow chart indicates intermediate data in running of the flow chart; and displaying, by the application interface layer, the running interface in response to a sixth operation on the running interface displaying control, to enable the user to view the intermediate data in running of the flow chart based on the running interface.

21. The method according to claim 20, wherein the method further comprises: displaying, by the application interface layer, a running interface configuration control, wherein the running interface configuration control indicates the displaying of a

running interface configuration interface which is configured to configure contents displayed on the running interface; displaying, by the application interface layer, the running interface configuration interface in response to a seventh operation on the running interface configuration control, wherein the running interface configuration interface comprises a plurality of running interface configuration sub-controls, and any one of the plurality of running interface configuration sub-controls corresponds to an open option; and displaying, by the application interface layer, a content source option in response to an eighth operation on the open option of a first running interface configuration sub-control of the plurality of the running interface configuration sub-controls, and binding, by the application interface layer, the first running interface configuration sub-control with one module in the flow chart in response to a ninth operation on the content source option; wherein, the running information displayed on the running interface comprises data of the module bound to the first running interface configuration sub-control.

22. The method according to claim 20, wherein the method further comprises: exporting, by the data processing layer, a program code of the running interface as a second code package; and controlling, by the data processing layer, the application interface layer to display a second code control corresponding to the second code package, wherein the second code control is configured to instruct the user to download the second code package, and the second code package is allowed to be imported into the third-party system and enables the running interface to be displayed after being run on the third-party system.

23. The method according to any one of claims 14 to 22, wherein the data processing layer comprises a data scheduling layer and a functional module layer; module data of the N modules is stored in the functional module layer; the calling and running, by the data processing layer, the module data of the modules respectively corresponding to the one or more module displaying controls in response to the flow chart displayed by the application interface layer, to obtain the data processing result of the flow chart, comprises: sequentially calling and running, by the data scheduling layer, the module data of the modules respectively corresponding to the one or more module displaying controls from the functional module layer according to the

execution order of the modules respectively corresponding to the one or more module displaying controls in the flow chart, to obtain the data processing result.

24. The method according to claim 14, wherein the method further comprises: displaying, by the application interface layer, a module registration control, and displaying, by the application interface layer, the N module displaying controls in response to the N modules selected by the user from M modules through the module registration control, wherein module data of the M modules is stored in the data processing layer, and M is a positive integer greater than or equal to N.

25. The method according to claim 24, wherein the N modules are modules verified by the data processing layer to be valid, and the modules verified to be valid refer to corresponding modules that the user has permission to use.

26. The method according to claim 14, wherein the method further comprises: displaying, by the application interface layer, a calling control for any one of the module displaying controls, wherein the calling control indicates that the user is allowed to call a module corresponding to the any one of the module displaying controls in a third-party system; displaying, by the application interface layer, a port link of module data of the module corresponding to the any one of the module displaying controls in response to a tenth operation on the calling control; and acquiring, by the application interface layer, the module data of the module corresponding to the any one of the module displaying controls from the data processing layer based on the port link when detecting a data calling request for the port link sent by the third-party system, and sending, by the application interface layer, the acquired module data to the third-party system.

27. An apparatus for visual software development, comprising: a processor; a memory for storing processor-executable instructions; wherein, the processor is configured to execute the method according to any one of claims 14 to 26.

28. A computer-readable storage medium having instructions stored thereon which, when executed by a processor, execute the method according to any one of claims 14 to 26.

29. A computer program product containing instructions which, when run on a computer, cause the computer to execute the method according to any one of claims 14 to 26.