

CLAIMS

1. A graphical development method applied to program development in SQL language, comprising:

encapsulating, in each DAG component, an operator for implementing a function of the DAG component, and a name of another DAG component allowed to be connected;

presenting a plurality of visualized DAG components to a user;

receiving selection of a plurality of DAG components by the user, and establishing a connectivity of the DAG components according to the function and the DAG component allowed to be connected of a respective DAG component, thereby generating a DAG component graph;

the method further comprising, after generating the DAG component graph:

analyzing code relationship of the DAG component graph and generating SQL codes;

receiving a switch instruction and performing an operation inverse to the analyzing to form the DAG component graph;

wherein the analyzing comprises:

resolving the connectivity formed by the respective DAG components and links into a data structure;

decomposing the data structure into a plurality of logical operators according to a statement configuration;

generating statement codes corresponding to the plurality of logical operators;

wherein

the encapsulating further comprises: encapsulating one or more parameter or attribute configuration item; and

wherein receiving selection of the plurality of DAG components by the user further comprises:

displaying a parameter or attribute configuration panel for the DAG component selected by the user;

receiving the configuration parameter or attribute data input by the user, and verifying the data in real time,

the method further comprising:

re-verifying, each time when the connectivity is changed by adding a DAG component into the connectivity, a downstream path of the added DAG component, wherein the downstream path includes the respective DAG components and the links formed thereby in the connectivity; or,

re-verifying, when any of the DAG components is clicked, links for the DAG components in a downstream path of the clicked DAG component.

2. The method of claim 1, wherein the re-verifying comprises: checking correctness of a field list and type of one or more input/output of each DAG component.

3. The method of claim 1, further comprising: selecting a region including a plurality of DAG components in the established connectivity, and saving the region as a callable template.

4. A computer-readable storage medium storing a computer program which, when executed by a processor, implements the method according to any one of claims 1 to 3.

5. A computer device incorporated with the computer storage medium of claim 4.

6. A graphical development apparatus comprising:

an encapsulation module configured to encapsulate, in each DAG component, an operator for implementing a function of a DAG component, and a name of another DAG component allowed to be connected;

a presentation module configured to present a plurality of visualized DAG components to a user;

a graphics module configured to receive selection of a plurality of DAG components by the user, establish a connectivity of the DAG components according to the function and the DAG component allowed to be connected of a respective DAG component, thereby generate a DAG component graph;

wherein the graphics module is coupled to an analysis module configured to analyze code relationship of the DAG component graph and generate SQL codes;

wherein the analysis module is coupled to a conversion module, which is configured to receive a switch instruction and perform an operation inverse to the analysis module to form the DAG component graph;

wherein the analysis module comprises:

a first sub-module configured to resolve the connectivity formed by the respective DAG components and links into a data structure;

a second sub-module configured to decompose the data structure into a plurality of logical operators according to a statement configuration; and

a third sub-module configured to generate statement codes correspondingly to the plurality of logical operators;

wherein the encapsulation module is further configured to encapsulate one or more parameter or attribute configuration item; and

wherein the graphics module is coupled to a parameter verification module, which is configured to display a parameter or attribute configuration panel for the DAG component selected

by the user, and receive the configuration parameter or attribute data input by the user and verify the data in real time;

wherein the graphics module is further coupled to a re-verification module, which is configured to

re-verify, each time when the connectivity is changed by adding a DAG component into the connectivity, a downstream path of the added DAG component, wherein the downstream path includes the respective DAG components and the links formed thereby in the connectivity; or

re-verify, when any of the DAG components is clicked, links for the DAG components in a downstream path of the clicked DAG component.

7. The apparatus of claim 6, wherein the graphics module is further coupled to a template module, which is configured to select a region including a plurality of DAG components in the established connectivity, and save it as a callable template.