[0084] Every menu object should be represented in the highest level object diagram since many menus often are available in most, if not all, windows of an application.

[0085] In an OED, an event script should be drawn linked to the object that invokes it rather than to the main object.

[0086] All method objects should be shown in an OED. If a method object appears in a window OED (i.e., the main object of the OED is a window), it means that the method is local to that window. On the other hand, if a method object is defined in the application/system OED, the method is global and can be used in any window. Unlike event scripts, it is advisable to connect method objects to the main object of the OED rather than the object that invokes it.

[0087] Record objects, general objects, data transfer objects, and database objects generally will not need to have their own diagram. Often, they may just be represented in the highest level object diagram.

[0088] Having thus described a few particular embodiments of the invention, various alterations, modifications, and improvements will readily occur to those skilled in the art. Such alterations, modifications and improvements as are made obvious by this disclosure are intended to be part of this description though not expressly stated herein, and are intended to be within the spirit and scope of the invention. Accordingly, the foregoing description is by way of example only, and not limiting. The invention is limited only as defined in the following claims and equivalents thereto.

What is claimed is:

1. A method for graphically representing object oriented programming logic, the method comprising the steps of:

(1) providing a plurality of different symbols for use in a diagram of object oriented programming logic, each different symbol representing a different type of object;

(2) selecting an object as a main object of the logic to be represented in the diagram;

(3) drawing a symbol corresponding to the main object and labeling the symbol with a label descriptive of the object's features so that it is distinguishable from other symbols of the same object type;

(4) for each object assigned to or defined within the main object, drawing a symbol corresponding to that object and labeling the symbol with a label descriptive of the object's features; and

(5) drawing a line between each object drawn in step (4) and another object in the graphical representation to which it is assigned or within which it is defined.

2. The method of claim 1 further comprising the step of:

(6) providing a plurality of additional symbols for use in the diagram, each of the additional symbols representing an object oriented programming element other than an object.

3. The method of claim 1 further comprising the step of:

(7) graphically denoting the main object in the diagram by drawing another symbol around the symbol for the main object.

4. The method of claim 3 wherein step (6) comprises drawing a circle completely enclosing the symbol of the main object.

5. The method of claim 1 wherein the labels comprise text.

6. The method of claim 1 wherein step (5) comprises drawing the line between the object defined in step (4) and another object it is most directly assigned to or is most directly defined within.

7. The method of claim 1 wherein the method is used to document software.

8. The method of claim 1 wherein the method is used to prepare a program specification.

9. The method of claim 1 further comprising the step of:

(8) repeating steps (1)-(5) to prepare a plurality of separate diagrams corresponding to separate parts of an overall application and wherein a first object is the main object appearing in at least a first one of the diagrams and is not a main object appearing in at least a second one of the diagrams.

10. The method of claim 9 wherein the second diagram does not disclose objects assigned to and defined within the first object and the first diagram does disclose objects assigned to and defined within the first object.

11. The method of claim 10 wherein the second diagram is an application-level representation disclosing an overall software system.

12. The method of claim 10 wherein the label for the first object in the second diagram identifies the first diagram as disclosing further details of the first object.

13. The method of claim 1 wherein the symbols representing different object types include:

a first symbol for representing objects that are application type objects;

a second symbol for representing objects that are window type objects;

a third symbol for representing objects that are class type objects;

a fourth symbol for representing objects that are event script type objects; and

a fifth symbol for representing objects that are method type objects.

14. The method of claim 1 wherein the symbols representing different object types include:

a first symbol for representing objects that are application type objects;

a second symbol for representing objects that are window type objects;

a third symbol for representing objects that are class type objects;

a fourth symbol for representing objects that are event script type objects; and

a fifth symbol for representing objects that are method type objects.

and wherein the symbols representing additional program elements include:

a sixth symbol for representing data transfer;

a seventh symbol for representing databases;

an eighth symbol for representing remote links; and