

METHOD AND SYSTEM FOR OPTIMIZING STATIC AND DYNAMIC BINDING OF PRESENTATION OBJECTS WITH THE OBJECT DATA THEY REPRESENT

BACKGROUND OF THE INVENTION

1. Technical Field

The present invention is related generally to an improved data processing system, and in particular to a method and system for binding objects. Still more particularly, the present invention is related to a method in a system for binding Graphical User Interface objects Presentation objects and application objects.

2. Description of the Related Art

The development of application and system software for data processing systems has traditionally been a time consuming task. The field of software engineering has attempted to overcome the limitations of traditional techniques by proposing new, more efficient software development models. Object oriented programming has emerged as a promising technology that will allow rapid development, implementation and customization of new system software. Each object has certain data attributes and processes or methods that operate on that data. Data is said to be "encapsulated" by an object and can only be modified by the object methods are invoked by sending a message to an object identifying the method and supplying any needed arguments.

Object oriented systems have two important properties in addition to encapsulation. "Inheritance" is the ability to derive a new object from an existing object and inherit all properties, including methods and data structure, from the existing object. The new object may have certain unique features which are supplied as overrides or modifications to the existing class. I.e. a new subclass needs to specify only the functions and data that distinguish that class from the existing more general class.

The ability to override an existing method description enables polymorphism, the ability to have a single message to an object be processed in different ways depending on the object itself.

Inheritance and polymorphism create a powerful structure for implementing new software systems. The software developer does not have to develop each piece of a system, he or she need only specify the unique features of the system.

The power of object oriented systems is realized through the development of system "frameworks." A framework is a collection of base classes that can be used by a system implementor to create a final systems product. The framework is defined and developed to work together as a system. Conceptually, the framework is much like a set of standard hardware components used by computer hardware builders. Each of the components has certain defined functions and interfaces and the engineer assembles these components according to a particular design to create a unique hardware system.

One object oriented programming system is the System Object Model (SOM). More information on SOM can be found in SOM objects developer tool kit users guide, version 2.0, June 1993, available from international business machines corporations.

With presently available Graphic User Interface (GUI) applications, many GUI library and tools are available for user independent development. Some are server type tools,

others are class libraries. Every application, however, which uses those tools must still provide the code to use those methods or functions.

The GUI tools typically provide some of the necessary code to present the graphic objects (i.e. icon, folder, action bar, entry field, lists, etc.) on the screen. The coding and logical flow of the invocation of those methods/functions as well as special logic for the connections between the GUI objects and the application objects were left to the developer. A GUI object is an object that is visible to a user. An application object is an object that is not visible. Typically, a binding is created between a GUI object and an application object to allow interaction between a user and the application object using a GUI.

It is desirable to be able to predefine the binding between the GUI objects and the application objects in a generic fashion and to relieve the application object from the necessity to write all the required code, which is typically the same for every window, for each control type, across the system, lending to a great deal of redundancy and inefficiency. These features are specifically desirable in an object oriented environment where GUI objects and application objects are present and each one has an attribute which can be defined and addressed.

Therefore, it would be advantageous to have a method and system for providing binding between GUI objects and application objects in a generic way and to relieve the application from the necessity from providing all the required code.

SUMMARY OF THE INVENTION

It is one object of the present invention to provide an improved data processing system.

It is another object of the present invention to provide a method and system for binding objects.

It is yet another object of the present invention to provide a system for binding Graphical User Interface objects Presentation objects and application objects.

The present invention provides a data processing system having an object oriented environment, wherein the object oriented environment includes a plurality of graphic user interface objects and a plurality of application objects, each graphic user object having a unique identifier and a selected attribute and each application object having a preselected response. Graphic user interface objects associated with an application object are identified. A table is created specifying relationships between the graphic user interface objects and each graphic user interface object and the application object. Each time the application object is initialized, the table is used to specify objects within the application object.

The above as well as additional objectives, features, and advantages of the present invention will become apparent in the following detailed written description.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, as well as a preferred mode of use, further objectives and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein: