

**SYSTEM AND METHOD FOR
CONTROLLING VERSIONS OF OBJECTS IN
AN OBJECT ORIENTED COMPUTING
ENVIRONMENT**

FIELD OF THE INVENTION

This invention relates to data processing systems and methods and more particularly to object oriented computing environments.

BACKGROUND OF THE INVENTION

Object oriented programming systems and processes, also referred to as "object oriented computing environments", have been the subject of much investigation and interest in state of the art data processing environments. As is well known to those having skill in the art, object oriented programming systems are composed of a large number of "objects". An object is a data structure, also referred to as a "frame", and a set of operations or functions, also referred to as "methods", that can access that data structure. The frame has many "slots", each of which contains an "attribute" of the data in the slot. The attribute may be a primitive (such as an integer or string) or an object reference which is a pointer to another object. Objects having identical data structures and common behavior can be grouped together into, and collectively identified as, a "class".

Each defined class of objects will usually be manifested in a number of "instances". Each instance contains the particular data structure for a particular example of the object. In an object oriented computing environment, the data is processed by requesting an object to perform one of its methods by sending the object a "message". The receiving object responds to the message by choosing the method that implements the message name, executing this method on the named instance, and returning control to the calling high level routine along with the results of the method. The relationships between classes, objects and instances are established during "build time" or generation of the object oriented computing environment, i.e. prior to "run time" or execution of the object oriented computing environment.

In addition to the relationships between classes, objects and instances identified above, inheritance relationships also exist between two or more classes such that a first class may be considered a "parent" of a second class and the second class may be considered a "child" of the first class. In other words, the first class is an ancestor of the second class and the second class is a descendant of the first class, such that the second class (i.e., the descendant) is said to inherit from the first class (i.e. the ancestor). The data structure of the child class includes all of the attributes of the parent class.

Object oriented systems have heretofore recognized "versions" of objects. A version of an object is the same data as the object at a different point in time. For example, an object which relates to a "work in progress", is a separate version of the same object data which relates to a completed and approved work. Many applications also require historical records of data as it existed at various points in time. Thus, different versions of an object are required.

Heretofore, versions of an object were treated as totally independent objects. Complicated processing was required to implement a version control mechanism. Other applications have recognized versions of objects and have used customized logic to implement version control. In this case, the version control system of one application was not usable

with other applications, so that application portability was reduced.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide systems and methods for controlling versions of objects in an object oriented computing environment.

It is another object of the invention to provide systems and methods for generically controlling versions of objects in an object oriented computing environment to thereby reduce the amount of customized logic which must be generated for version control.

These and other objects are provided, according to the present invention, by a system and method for controlling versions of selected objects in an object oriented computing system on a computing platform, wherein each object includes an object frame containing data attributes and at least one object method for performing actions on the associated objects. The version control system creates a plurality of versions of a selected object. The object frame of each version includes unique data attributes and a common logical key attribute which is used to identify all versions of an object. The object frame of each version also includes a unique combination of an insert sequence attribute and an extract sequence attribute. The unique combination of insert sequence attribute and extract sequence attribute allows selection of a unique version of an object based on time.

According to the invention, the plurality of versions of the object are identified based on the common logical key attribute associated therewith. In order to select one or more versions of the identified object, a version identifier, such as a version time, is converted into a selection sequence value. Each application which uses the version controlled objects may include its own logic for converting a version identifier such as time into a selection sequence value, based on the unique requirements of that application.

The selection sequence value is then compared to the insert sequence attributes and extract sequence attributes of the versions of the selected object in order to select at least one of the versions of the selected object. The comparison may find an applicable version having an insert sequence attribute which is less than or equal to the selection sequence value and an extract sequence attribute which is greater than the selection sequence value. This selection method finds one applicable version valid at a particular time. Alternatively, all versions which were inserted by a certain time can be found by finding those versions having an insert sequence attribute which is equal to the selection sequence value. As another alternative, all versions which were extracted by a particular time can be found by identifying those versions having an extract sequence attribute which is equal to the selection sequence value.

According to another aspect of the invention, the object frame of each version also includes a perspective attribute for providing another selection criteria for selecting one or more versions of an object. When identifying versions, a perspective value can be compared to the perspective attributes of all of the versions, and only those versions which have a perspective attribute equal to the perspective value can be selected. This comparison can be made in addition to the comparison by insert sequence attribute and extract sequence! attribute.

The invention also supports updates to version controlled objects. Thus, new versions can be added, existing versions