cess for the composite data of the character, graphic and sign data.

As set forth above, according to the present invention, since the version management can be performed effectively using the multi-window, the efficiency for edition of versions can be remarkably improved. Also, by utilizing the common buffer for a plurality of windows, the required capacity of the buffer can be reduced. Furthermore, by performing editing for the newly derived version utilizing the window operation history between the versions, redundant entry of an identical editing operation can be avoided for facilitating the editing process.

I claim:

1. A version management method for performing management of a plurality of versions of a data file in a multi-window environment, comprising the steps of:

(a) displaying contents of the versions of the data file in windows on a display screen, the windows including a basic window for a major version and at least one derived window for at least one minor version derived from a parent window;

(b) retaining an identifier of the parent window and a differential operation history from the parent window with regard to each derived window in an operation history management table, the differential operation history detailing what and how changes were made in contents between the minor version corresponding to each derived window and a parent version corresponding to the parent window; and

(c) automatically updating corresponding contents of the operation history management table, when a new window for a new version is created as a derivative of one of the windows on the display screen and when an editing operation is performed in one of the windows, to modify the contents of the minor version corresponding thereto.

2. A version management method as claimed in claim 1, further comprising the step of (d) storing the contents of the major and minor versions of the data file in respective buffer memories.

3. A version management method as claimed in claim 2,

wherein the at least one minor version includes a plurality of minor versions corresponding to a plurality of derived windows, and

wherein said version management method further comprises the steps of:

(e) generating from a first window as the parent window a second window as one of the derived windows;

(f) editing the contents of a first minor version in the second window; and

(g) replacing the major version with one of the minor versions as a new major version.

4. A version management method as claimed in claim 3, wherein said generating in step (e) comprises the substeps of:

(e1) opening the second window on the display screen as one of the derived windows;

(e2) copying the contents of the buffer memory corresponding to the first window to the buffer memory corresponding to the second window;

(e3) redrawing the contents of the buffer memory in the second window; and

(e4) storing an identifier of the first window in the operation history management table as the parent window.

5. A version management method as claimed in claim 3, wherein said editing in step (f) comprises the substeps of:

(f1) editing the contents of the buffer memory corresponding to the second window according to the editing operation; and

(f2) adding data to the operation history management table representing the editing operation as the differential operation history of the second window.

6. A version management method as claimed in claim 3, wherein said replacing in step (g) comprises the substeps of:

(g1) selecting a third window to become a new basic window for a new major version from among the derived windows;

(g2) obtaining a sequence of parent windows of the third window using identifiers of the parent windows in the operation history management table;

(g3) applying operations corresponding to the differential operation histories of the parent windows and the third window to the buffer memory corresponding to the basic window; and

(g4) erasing the contents of the operation history management table and the contents of the buffer memories corresponding to all of the derived windows.

7. A version management method as claimed in claim 3, wherein said replacing in step (g) comprises the substeps of:

(g1) selecting a third window to become a new basic window for a new major version from among the derived windows;

(g2) copying the contents of the buffer memory corresponding to the third window to the buffer memory corresponding to the basic window; and

(g3) erasing the contents of the operation history management table and the contents of the buffer memories corresponding to all of the derived windows.

8. A version management method as claimed in claim 3, wherein said editing in step (f) comprises the substeps of:

(f1) obtaining one of a differential operation history and a sequence of differential operation histories from the first window to the second window;

(f2) applying operations corresponding to the one the differential operation history and the differential operation histories obtained in step (f1) to the buffer memory corresponding to a third window; and

(f3) adding the one of the differential operation history and the differential operation histories to obtain the operation history management table corresponding to said editing displayed in the third window.

9. A version management method as claimed in claim 1, further comprising the steps of:

(d) retaining the contents of one of the versions in a buffer memory; and

(e) retaining an identifier of a corresponding window for the one of the versions in a current window memory, to define the corresponding window as a current window.

10. A version management method as claimed in claim 9,