

with the stored documents of intermediate versions between the higher version and the lower version and the stored document of the higher version or the relevant information therein.

5. The method of claim 1, wherein the converting step comprises:

receiving a document of a lower version from an application of the lower version;

successively converting the document of the lower version into documents of intermediate versions between a higher version and the lower version and a document of the higher version by using at least one differentiation models between data models between the higher version and the lower version in the conversion stack, and storing the document of the lower version and the documents of intermediate versions between the higher version and the lower version or the relevant information therein;

providing the document of the higher version to an application of the higher version, so that required modifications to the document of the higher version is performed by the application of the higher version;

successively converting the modified document of the higher version into modified documents of intermediate versions and a modified document of the lower version by using one or more differentiation models between data models between the higher version and the lower version in the conversion stack, and by successively merging with the stored documents of intermediate versions between the higher version and the lower version and the document of the lower version or the relevant information therein; and

sending the modified document of the lower version to the application of the lower version.

6. The method of claim 1, wherein the conversion stack further comprises;

a verification model which contains rules for verifying whether a document conforms to a data model, and the conversion further comprises verifying a converted document according to the verifying module, so as to make the converted document conform to the data model of the version as the conversion target.

7. The method of claim 1, wherein the conversion stack is in an application of each version and includes differentiation models between data models of the version and lower versions, and the conversion is executed in the application of the version.

8. The method of claim 1, wherein the conversion stack is in a network server and the conversion is executed as a networked service in the network server.

9. The method of claim 1, wherein providing the conversion stack further comprises:

obtaining the data models of different versions of the application; and

generating the differentiation models between data models of different versions of the application according to the data models of the different versions of the application, and storing the differentiation models into the conversion stack.

10. An apparatus for providing inter-version document compatibility, comprising:

a conversion stack which includes differentiation models between data models of different versions of an application; and

a conversion module configured for converting between documents of different versions of the application by using the conversion stack, so as to provide compatibility between the documents of the different versions of the application.

11. The apparatus of claim 10, wherein differentiation modes between data models of different versions of the application are differentiation models between data models of adjacent versions of the application.

12. The apparatus of claim 10, wherein the conversion module is further configured for:

obtaining a document of a first version;

successively converting the document of the first version into documents of intermediate versions between the first version and a second version and a document of the second version by using one or more differentiation models between data models between the first version and the second version in the conversion stack; and

providing the document of the second version.

13. The apparatus according to claim 10, wherein the conversion module is further configured for:

obtaining a document of a higher version;

successively converting the document of the higher version into documents of intermediate versions between the higher version and a lower version and a document of the lower version by using one or more differentiation models between data models between the higher version and the lower version in the conversion stack, and storing the document of the higher version and the documents of intermediate versions between the higher version and the lower version or the relevant information therein;

sending the document of the lower version to an application of the lower version so that required modifications to the document of the lower version may be performed by the application of the lower version;

receiving a modified document of the lower version from the application of the lower version; and

successively converting the modified document of the lower version into modified documents of intermediate versions and a modified document of the higher version by using at least one differentiation model between data models between the higher version and the lower version in the conversion stack, and by successively merging with the at least one stored document of intermediate versions between the higher version and the lower version and the stored document of the higher version or the relevant information therein through a merging module.

14. The apparatus of claim 13, further comprising:

a merging module configured for successively merging at least one incomplete document of intermediate versions and an incomplete document of the higher version successively converted from the modified document of the lower version with the at least one stored document of intermediate versions between the higher version and the lower version and the stored document of the higher version or the relevant information therein, to generate the at least one modified document of intermediate versions, and the modified document of the higher version.

15. The apparatus of claim 10, wherein the conversion module is further configured for:

receiving a document of a lower version from an application of the lower version;

successively converting the document of the lower version into documents of intermediate versions between a