(11) Application No. AU 2005225110 A1

(19) AUSTRALIAN PATENT OFFICE

(54) Title
Method and system for linking data ranges of a computer-generated document with associated Extensible Markup Language elements

(51)[6] International Patent Classification(s)
G06F 17/30 (2006.01) 20060101AFI2006010
G06F 17/30 1BHAU

(21) Application No: 2005225110 (22) Application Date: 2005.10.21

(30) Priority Data

(31) Number	(32) Date	(33) Country
11018569	2004.12.20	US

(43) Publication Date : 2006.07.06
(43) Publication Journal Date : 2006.07.06

(71) Applicant(s)
Microsoft Corporation

(72) Inventor(s)
Little, Robert; Villaron, Shawn Alan; Jones, Brian

(74) Agent/Attorney
Davies Collison Cave, 1 Nicholson Street, MELBOURNE, VIC, 3000

Abstract

Methods and systems are provided for separating the data of a structured document from markup structure applied to the document and for linking the separately maintained data with associated markup structure applied to the document. The data stream in an Extensible Markup Language (XML) represented document is moved outside of the document structure into a data file that is linked to the document structure. The data file and the XML-represented document structure are maintained in association with a single file (e.g., document), but in separate object models. Links are established between XML elements applied to the document and data ranges of the data file. Each of the XML structure file and the data file may be accessed and edited separately without affecting the structure or data of the other file until the two files are merged for presenting the associated document.

2005225110 21 Oct 2005

AUSTRALIA

PATENTS ACT 1990

COMPLETE SPECIFICATION

NAME OF APPLICANT(S)::

Microsoft Corporation

ADDRESS FOR SERVICE:

DAVIES COLLISON CAVE
Patent Attorneys
1 Nicholson Street, Melbourne, 3000, Australia

INVENTION TITLE:

Method and system for linking data ranges of a computer-generated document with associated Extensible Markup Language elements

The following statement is a full description of this invention, including the best method of performing it known to me/us:-

5102

Field of the Invention

The present invention generally relates to managing data in computer-generated documents. More particularly, the present invention relates to methods and systems for linking data ranges of a data file with associated Extensible Markup Language elements of a computer-generated document.

Background of the Invention

With the advent of the computer age, computer and software users have grown accustomed to user-friendly software applications that help then write, calculate, organize, prepare presentations, send and receive electronic mail, make music, and the like. For example, modern electronic word processing applications allow users to prepare a variety of useful documents. Modern spreadsheet applications allow users to enter, manipulate, and organize data. Modern electronic slide presentation applications allow users to create a variety of slide presentations containing text, pictures, data or other useful objects.

Computer-generated documents, for example, word processing documents, may be structured and formatted according to a markup language such as the Extensible Markup Language (XML). Applying XML structure to such documents, or representing the entire document as an XML file, provides many advantages. For example, a given text or data range in a template document may be structured for containing data of a certain type (e.g., date, heading, conclusion, summary, etc.). Additionally, by representing the entire document as XML, all the application information used to create that document is available in a text format for external sources to parse/edit. Subsequently, a consuming application may readily parse the document to obtain and use either the desired text, formatting information, structural information, or even just

the data from the document according to the structure of the actual persistent XML saved out with the document.

Unfortunately, editing behaviors of such structured documents are often fragile because the documents are limited by the fact that the positioning of XML tags (markup) on the document surface determines the structure of the XML instance document in an associated user-defined schema file. Problems often arise from common user operations (e.g., copy/paste from one section of a document to another section) may corrupt the XML structure applied to the document. Additionally, all elements from a user-defined schema file must be included in some form on the document surface. This makes it impossible to choose only a subset of the data to present to the user for editing while maintaining the rest for other uses (workflow; searching; management; etc.). Other problems include an inability for solution creators to use a schema file as a method for carrying true metadata about the document. Moreover, elements that are semantically unnecessary on the document surface (e.g., non-leaf elements which are not marking up mixed content) must be included which further increase the fragility associated with common user operations. The final difficulty for a solution creator is that in order to deal with the data they care about, they must also navigate through all the application specific markup that is not as interesting to them. They must also make sure that when editing values of their own custom data, they also need to maintain the proper application markup.

It is with respect to these and other considerations that the present invention has been made.

Summary of the Invention

Embodiments of the present invention solve the above and other problems by providing methods and systems for separating the custom data of a structured document from application markup structure applied for the actual representation of the document and for linking the separately maintained data with associated markup structure applied to the document. According to embodiments of the present invention, the data stream in an Extensible Markup Language (XML) represented document is moved outside of the

document structure into a data file that is linked to the document structure. The data file and the XML-represented document structure are maintained in association with a single file (e.g., document), but in separate parallel object models.

Links are established between XML elements applied to the document and data ranges of the data file. Accordingly, a document author/editor may markup the document with XML structure that is linked to data that is associated with structural elements of the document. An end user of the document may access the XML structure of the document for editing the XML structure of the document without corrupting the data of the file. Likewise, an end user may access the separately maintained data file to edit data without corrupting the XML structure of the document. When the document is opened, the XML structure and the associated data are merged for presentation to a user such that the data is presented in the document according to the structure applied to the document. This allows for solutions to be built against the custom data without the need to understand any of the application specific markup (as they are kept separate). This is true when editing the data while the file is loaded in the application, as well as when the file is saved and the solution is acting on the file itself.

These and other features and advantages, which characterize the present invention, will be apparent from a reading of the following detailed description and a review of the associated drawings. It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention as claimed.

Brief Description of the Drawings

Fig. 1 illustrates an exemplary computing operating environment for embodiments of the present invention.

Fig. 2 illustrates a computer-generated document comprising an Extensible Markup Language structure file and an associated data file according to embodiments of the present invention.

Detailed Description

As briefly described above, embodiments of the present invention are directed to methods and systems for separating the data of a structured document from the markup structure applied to the document. The separately maintained data is linked with associated markup structure applied to the document. Accordingly, an end user of the document may access the structure of the document for editing the structure without corrupting the data of the file. Likewise, an end user may access the separately maintained data file to edit data without corrupting the structure of the document. When the document is opened, the structure and the associated data are merged for presentation to a user such that the data is presented in the document according to the structure applied to the document. These embodiments may be combined, other embodiments may be utilized, and structural changes may be made without departing from the spirit or scope of the present invention. The following detailed description is therefore not to be taken in a limiting sense and the scope of the present invention is defined by the appended claims and their equivalents.

Referring now to the drawings, in which like numerals refer to like elements through the several figures, aspects of the present invention and an exemplary operating environment will be described. Fig. 1 and the following discussion are intended to provide a brief, general description of a suitable computing environment in which the invention may be implemented. While the invention will be described in the general context of program modules that execute in conjunction with an application program that runs on an operating system on a personal computer, those skilled in the art will recognize that the invention may also be implemented in combination with other program modules.

Generally, program modules include routines, programs, components, data structures, and other types of structures that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are

performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

Turning now to Fig. 1, an illustrative architecture for a personal computer 2 for practicing the various embodiments of the invention will be described. The computer architecture shown in Fig. 1 illustrates a conventional personal computer, including a central processing unit 4 ("CPU"), a system memory 6, including a random access memory 8 ("RAM") and a read-only memory ("ROM") 10, and a system bus 12 that couples the memory to the CPU 4. A basic input/output system containing the basic routines that help to transfer information between elements within the computer, such as during startup, is stored in the ROM 10. The personal computer 2 further includes a mass storage device 14 for storing an operating system 16, application programs, such as the application program 205, and data.

The mass storage device 14 is connected to the CPU 4 through a mass storage controller (not shown) connected to the bus 12. The mass storage device 14 and its associated computer-readable media, provide non-volatile storage for the personal computer 2. Although the description of computer-readable media contained herein refers to a mass storage device, such as a hard disk or CD-ROM drive, it should be appreciated by those skilled in the art that computer-readable media can be any available media that can be accessed by the personal computer 2.

By way of example, and not limitation, computer-readable media may comprise computer storage media and communication media. Computer storage media includes volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EPROM, EEPROM, flash memory or other solid state memory technology, CD-ROM, DVD, or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by the computer.

According to various embodiments of the invention, the personal computer 2 may operate in a networked environment using logical connections to remote computers through a TCP/IP network 18, such as the Internet. The personal computer 2 may connect to the TCP/IP network 18 through a network interface unit 20 connected to the bus 12. It should be appreciated that the network interface unit 20 may also be utilized to connect to other types of networks and remote computer systems. The personal computer 2 may also include an input/output controller 22 for receiving and processing input from a number of devices, including a keyboard or mouse (not shown). Similarly, an input/output controller 22 may provide output to a display screen, a printer, or other type of output device.

As mentioned briefly above, a number of program modules and data files may be stored in the mass storage device 14 and RAM 8 of the personal computer 2, including an operating system 16 suitable for controlling the operation of a networked personal computer, such as the WINDOWS operating systems from Microsoft Corporation of Redmond, Washington. The mass storage device 14 and RAM 8 may also store one or more application programs. In particular, the mass storage device 14 and RAM 8 may store an application program 205 for providing a variety of functionalities to a user. For instance, the application program 205 may comprise many types of programs such as a word processing application, a spreadsheet application, a desktop publishing application, and the like. According to an embodiment of the present invention, the application program 205 comprises a multiple functionality software application suite for providing functionality from a number of different software applications. Some of the individual program modules that may comprise the multiple application suite 205 include a word processing application 125, a slide presentation application 135, a spreadsheet application 140 and a database application 145. An example of such a multiple functionality application suite 205 is OFFICE manufactured by Microsoft Corporation. Other software applications illustrated in Fig. 1 include an electronic mail application 130.

Fig. 2 illustrates a computer-generated document comprising an Extensible Markup Language structure file and an associated data file according to embodiments of

the present invention. Referring to Fig. 2, a computer-generated file 210 is illustrated, which is generated and/or edited by an application 205. As described above, the application 205 may be a single software application, such as a word processing application, a spreadsheet application, a slide presentation application, and the like. Or, the application 205 may represent a multiple application suite which includes multiple applications, for example, a word processing application, a spreadsheet application, a slide presentation application, and the like.

The file or document 210, for example, a word processing document, such as an article or memorandum, is comprised of two parallel, but separate, files 215, 220. According to embodiments of the present invention, XML structure applied to the file 210 according to an associated XML schema file 207 is stored in the XML structure file 215 under a document object model established for the XML structure file. For example, referring to the XML structure file 215, a first XML structural element 225 is applied to the structure file for an example heading section of the document 210. A second XML structural element 230 is applied to the XML structure file 215 for an example body section of the document. A third XML structural element 240 is applied to the XML structure file 215 for an example conclusion section of the document 210.

The schema file 207 is illustrative of an XML file associated with the file 210 for defining the XML structure applied to the file 210. For example, the schema file 207 may be used for defining the names and definitions of each of the XML elements 225, 230, 240 applied to the XML structure file 215. Similarly, the schema file 207 may be used for defining the data types and data properties that may be entered into the associated data file 220 for populating the overall file 210 according to the XML structure applied to the file 210 in the XML structure file 215.

According to embodiments of the present invention, rather than entering data for each of the structured sections of the file 210 into a single document along with the associated XML structure, the data for the file 210 is maintained in a separate data file 220 established under a separate, but parallel, document object model. As illustrated in Fig. 2, the example data file 220 contains a first data range 265 associated with the XML element 225 in the XML structure file 215. A second data range 275 is

associated with both the XML element 240. Links may be written between a single data range in the data file 220 and more than one associated XML structural element in the XML structure file 215. According to embodiments of the invention, all XML structural elements do not point to or are not linked to corresponding data ranges. Referring to Fig. 2, for example, the <body> element 230 and the <paragraph> element 235 are not linked to data ranges in the data file 220. Those elements may have been applied to the document for presentation and layout. Also, it may be that data supplied to the linked data ranges may come from a third party, but data for the <body> and <paragraph> sections may come from the document editor, and there may be no need to link those elements to a data source.

Each XML element in the XML structure file 215 is linked to an associated data range in the data file 220 via a data link. According to one implementation, the data link is an XPath. As is known to those skilled in the art, the XPath provides a means for linking an XML structural element in the XML file 215 with corresponding data ranges in the data file 220. According to an embodiment, the mapping between portions of the document and associated data ranges is not a direct mapping between an XML element and an associated data range as illustrated in Fig. 2. Instead, the mapping is between a "data binding" tag in the document that is associated with a desired portion of the document, and the "data binding" tag has an XPath as a value. As should be understood by those skilled in the art, other suitable means may be utilized for establishing a data link between a given XML structural element in the XML structure file with corresponding data ranges in the associated data file 220. For a detailed discussion of linking structured portions of a document to associated data ranges in a data file or database, see United States Patent Application, Serial No. 10/164,260 filed June 5, 2002, entitled "Mechanism for Downloading Software Components from a Remote Source for Use by a Local Software Application", which is incorporated herein by reference as if fully set out herein, and United States Patent Application, Serial No. 10/366,141, filed February 13, 2003, entitled "Linking Elements of a Document to Corresponding Fields, Queries and/or Procedures in a Database", which is incorporated herein by reference as if fully set out herein.

According to embodiments of the present invention, the document 210 serves as a container for the XML structure file 215 and the associated data file 220. As should be understood by those skilled in the art, the XML structure file 215 and the associated data file 220 may be maintained in a common memory location on the hard drive of a local computing device, or on a volatile memory source such as a floppy disk or CD. Alternatively, the XML structure file 215 and the associated data file 220 may be stored in different locations and may be linked to each other via a distributed computing network. For example, the data file 220 may be stored on a computer server located a distance from the separately stored XML structure file 215, and the link between given XML structure elements in the XML structure file 215 and associated data ranges in the data file 220 may be across a distributed computing network such as a local or wide area intranet or the Internet.

Referring still to Fig. 2, when the file 210 is opened via the application 205, the file may be opened as a single document showing data from the data file 220 presented in the document structured according to the XML structure file 215. For example, opening the file 210 may result in the display of a single document having a heading with heading data from the data range 265, having a body with body data from the data range 270 and having a conclusion including conclusion data from the data range 275. When the file 210 is thus opened, the separate XML structure file 215 and the associated data file 220 are synchronized to form and display the single document having the data from the data file 220 formatted and structured according to the XML elements applied to the document via the XML structure file 215.

Alternatively, if it is desired by an author or editor of the file 210 to affect editing changes, additions or deletions to the XML elements applied to the document via the XML structure file 215, the XML structure file 215 may be opened as a separate file without population of the XML structure with data from the associated data file 220. Thus, the author and/or editor may make XML structural or formatting changes to the XML structure file without being encumbered by the inclusion of data associated with any of the XML elements. For example, an author/editor of the file 210 may decide to add a new section to the example document illustrated at Fig. 2. For example,

the author/editor may decide to add a "summary" section to the file 210. For adding the summary section, the author/editor may add XML tags associated with a summary section to the XML structure file 215. After the XML structure is added to the XML structure file 215, a link is established between the new XML element(s) with an associated data range in the associated data file 220 if it is desired that data from a data range be associated with the new XML structure. That is, as described above, it is not necessary that all portions of the document be linked to associated data ranges. Some markup may be applied for presentation or formatting only.

According to embodiments of the present invention, an author/editor of the file 210 may similarly open the data file 220 separate from the XML structure file 215 for editing individual data ranges contained in the data file 220. For example, if the author/editor of the file 210 desires to edit the conclusion of the example word processing document/file 210, the author/editor may open the data file 220 and make changes to the data contained in the data range 275 illustrated in Fig. 2. Once one or more data ranges in the data file 220 are edited, the data file 220 may be saved without affecting any of the structural markup applied to the XML structure file 215. Subsequently, when the file 210 is opened by the application 205, the edited data applied to the data file 220, as described above, is displayed in the file 210 according to the associated XML structure linked to the edited data from the XML structure file 215.

According to embodiments of the present invention, a data editing application 208 may be utilized by an author/editor of the file 210 or by third parties for accessing the data file 220 and for editing data ranges contained in the data file 220. That is, the third party data editing application 208 may be a separate application from the application 205 with which the file 210 is created and/or edited. This is particularly advantageous for allowing third parties to prepare programs for running against the data file 220 on a periodic or automated basis. For example, a third-party accounting firm may be hired by a sales company to tabulate sales figures on a periodic basis. The third-party accounting firm may, on a periodic basis, open the data file 220 using a third-party data editing application 208 for automatically updating a data range contained in the data file associated with sales figures. Accordingly, when a sales

manager subsequently opens the file 210 with the application 205, the opened file 210 will show the updated sales figures applied to the data file 220, but shown according to the XML structure applied to the file by the XML structure file 215.

Advantageously, the third-party accounting firm, in this example, is able to affect data changes without the potential of changing or harming XML structure applied to the document via the XML structure file 215. Indeed, the third party does not need to know anything about the XML structure applied to the document. The third party can be completely ignorant of how the document will be structured so that the third party is only responsible for its own custom data that is populated into a data range for access by the document. In addition, for security purposes, it may be desired that the third-party accounting firm, in this example, may only access certain data ranges in the data file 220 because other data ranges in the data file may contain information that is confidential and that should not be accessible by the third-party accounting firm. According to embodiments of the present invention, the third-party accounting firm may make changes directly to the one or more data ranges to which it has access for updating the overall file 210 without having access to other data ranges in the data file 220 and without having access to the XML structure applied to the file 210 by the author/editor or owner of the file 210.

According to embodiments of the present invention, as data is changed in the file 210, live updates may be made to a corresponding data range contained in the data file 220. For example, if changes are made to data associated with the heading section 225, the application 205 may pass the data changes along the link 245 to the data range 265 for automatically changing the data contained in the data range 265. Alternatively, the changes made to data in the file 210 may be temporarily stored in a temporary data file that parallels the data file 220. Then, if the author/editor of the file 210 saves the file 210, the changes made to a section of the file 210 may be passed along the corresponding link to the data range in the data file 220 for persisting those changes in the data file 220.

Similarly, if an author/editor of the file 210 makes changes to the file 210 in a given section of the file 210 associated with XML structure and corresponding data

ranges, the author/editor may choose to "undo" the changes made to the data. According to one embodiment, if the changes made to the data are automatically persisted to associated data ranges in the data file 220, the application 205 may send data associated with the edited data range across an appropriate data link to the affected data range for replacing the data to a state prior to the change in the data by the author/editor. According to an alternate embodiment, if changes made to given sections of the file 210 are stored in a temporary data file until the file 210 is saved, an undo operation may be made by passing data to the temporary data file for restoring the affected data range in the temporary data file to a state of the data prior to the change made by the author/editor.

As briefly described herein, methods and systems are provided for separating markup structure applied to a document from corresponding data entered into the document to allow separate editing of the markup structure and associated data. Linking markup structure elements of the document with associated data ranges in a separate data file allow for the presentation of a merged document showing the entered data structured according to the applied markup elements. It will be apparent to those skilled in the art that various modifications or variations may be made in the present invention without departing from the scope or spirit of the invention. Other embodiments of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein.

Throughout this specification and the claims which follow, unless the context requires otherwise, the word "comprise", and variations such as "comprises" or "comprising", will be understood to imply the inclusion of a stated integer or step or group of integers or steps but not the exclusion of any other integer or step or group of integers or steps.

The reference to any prior art in this specification is not, and should not be taken as, an acknowledgment or any form of suggestion that that prior art forms part of the common general knowledge in Australia.

5

13

2005225110 21 Oct 2005

THE CLAIMS DEFINING THE INVENTION ARE AS FOLLOWS:

1. A method for managing markup structure and associated data in a computer-generated document, comprising:

applying Extensible Markup Language (XML) elements to a document;

generating an XML structure file for containing the XML elements applied to the document;

generating a data file for storing data entered into the document;

establishing a link from a given XML element contained in the XML structure file to a corresponding data range contained in the data file; and

upon launching the document, merging data contained in the data file with linked XML elements contained in the XML structure file for displaying the data contained in the data file in the document according to the XML elements applied to the document.

2. The method of Claim 1, further comprising providing an electronic file container for containing the XML structure file and the data file.

3. The method of Claim 2, whereby upon selection of the electronic file container, providing access to each of the XML structure file and the data file.

4. The method of Claim 2, further comprising:

allowing access to the XML structure file independent of the data file for editing the XML elements applied to the document ; and

allowing access to the data file independent of the XML structure file for editing data contained in the data file.

5. The method of Claim 1, further comprising:

receiving edits to a given data item displayed in the document; and

passing the edits to the given data item to the data file for persisting in an associated data range in the data file.

6. The method of Claim 5, further comprising passing the data edits across a link between an XML element associated with the edited data item to a corresponding data range in the data file.

7. A method for managing markup structure and associated data in a computer-generated document, comprising:

receiving an application of Extensible Markup Language (XML) elements to a document;

receiving an input of data into the document;

generating an XML structure file for containing the XML elements applied to the document;

generating a data file for storing data entered into the document;

separating the XML structure file and the data file into two separate files contained in an electronic container file;

establishing a link from a given XML element contained in the XML structure file to a corresponding data range in the data file; and

upon launching the document, merging data contained in the data file with linked XML elements contained in the XML structure file for displaying the data contained in the data file in the document according to a document structure dictated by the XML elements applied to the document.

8. The method of Claim 7, whereby upon selection of the electronic file container, providing access to each of the XML structure file and the data file.

9. The method of Claim 8, further comprising:

allowing access to the XML structure file independent of the data file for editing the XML elements applied to the document ; and

allowing access to the data file independent of the XML structure file for editing data contained in the data file.

10. The method of Claim 7, further comprising:

receiving edits to a given data item displayed in the document; and

passing the edits to the given data item to the data file for persisting in an associated data range in the data file.

11. The method of Claim 10, further comprising passing the data edits across a link between an XML element associated with the edited data item to a corresponding data range in the data file.

12. A computer-readable medium having stored thereon computer-executable instructions which when executed by a computer perform a method for managing markup structure and associated data in a computer-generated document, comprising:

applying Extensible Markup Language (XML) elements to a document;

generating an XML structure file for containing the XML elements applied to the document;

generating a data file for storing data entered into the document;

establishing a link from a given XML element contained in the XML structure file to a corresponding data range contained in the data file; and

upon launching the document, merging data contained in the data file with linked XML elements contained in the XML structure file for displaying the data contained in the data file in the document according to the XML elements applied to the document.

13. The computer-readable medium of Claim 12, further comprising providing an electronic file container for containing the XML structure file and the data file.

14. The computer-readable medium of Claim 13, whereby upon selection of the electronic file container, providing access to each of the XML structure file and the data file.

15. The computer-readable medium of Claim 13, further comprising allowing access to the XML structure file independent of the data file for editing the XML elements applied to the document.

16. The computer-readable medium of Claim 15, further comprising allowing access to the data file independent of the XML structure file for editing data contained in the data file.

17. The computer-readable medium of Claim 12, further comprising:

receiving edits to a given data item displayed in the document; and

passing the edits to the given data item to the data file for persisting in an associated data range in the data file.

18. The computer-readable medium of Claim 17, further comprising passing the data edits across a link between an XML element associated with the edited data item to a corresponding data range in the data file.

19. The computer-readable medium of Claim 12, whereby establishing a link from a given XML element contained in the XML structure file to a corresponding data range contained in the data file includes establishing an XPath from each XML element contained in the XML structure file that is linked to a corresponding data range contained in the data file.

20. The computer-readable medium of Claim 12, whereby one or more XML elements contained in the XML structure file are not linked to associated data ranges contained in the data file.

DATED this TWENTY FIRST day of OCTOBER 2005

Microsoft Corporation

by DAVIES COLLISON CAVE
Patent Attorneys for the applicant(s)

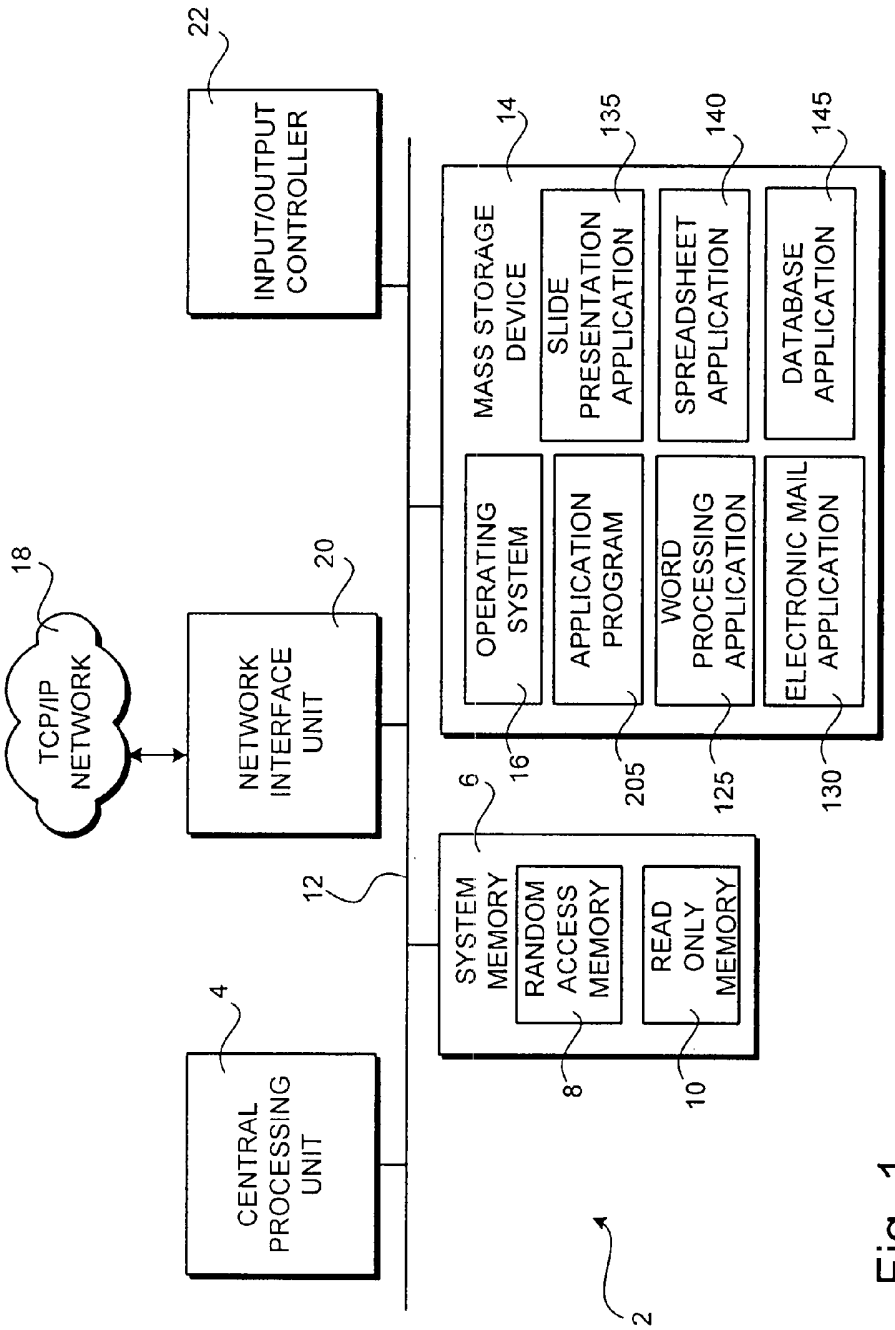
22
INPUT/OUTPUT CONTROLLER
14
MASS STORAGE DEVICE
135
SLIDE PRESENTATION APPLICATION
140
SPREADSHEET APPLICATION
145
DATABASE APPLICATION
OPERATING SYSTEM
APPLICATION PROGRAM
WORD PROCESSING APPLICATION
ELECTRONIC MAIL APPLICATION
16
205
125
130
18
TCP/IP NETWORK
20
NETWORK INTERFACE UNIT
6
SYSTEM MEMORY
RANDOM ACCESS MEMORY
READ ONLY MEMORY
8
10
12
4
CENTRAL PROCESSING UNIT
2

Fig. 1

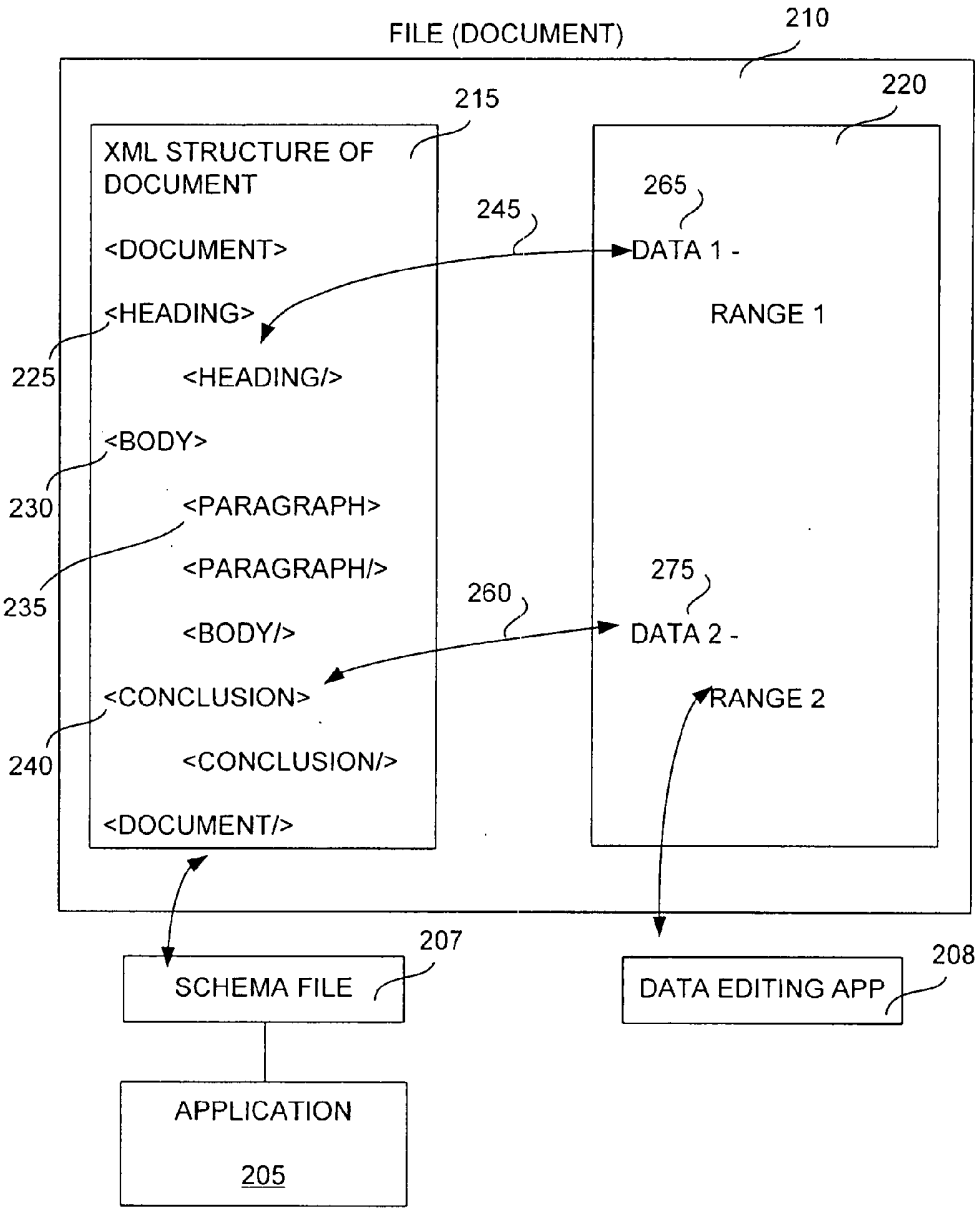
FILE (DOCUMENT)
210
215
220
XML STRUCTURE OF DOCUMENT
<DOCUMENT>
<HEADING>
<HEADING/>
<BODY>
<PARAGRAPH>
<PARAGRAPH/>
<BODY/>
<CONCLUSION>
<CONCLUSION/>
<DOCUMENT/>
245
265
DATA 1 -
RANGE 1
225
230
235
240
260
275
DATA 2 -
RANGE 2
207
208
SCHEMA FILE
DATA EDITING APP
APPLICATION
205

Fig. 2