(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 November 2004 (11.11.2004)

PCT

(10) International Publication Number
WO 2004/097675 A1

(51) International Patent Classification^7: G06F 17/30

(21) International Application Number: PCT/GB2004/001912

(22) International Filing Date: 4 May 2004 (04.05.2004)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0310218.3 2 May 2003 (02.05.2003) GB

(71) Applicants and
(72) Inventors: ROUSSEAU, David, Nicholas [GB/GB]; 30 Leich Close, Addlestone, Surrey KT15 1EL (GB). ROUSSEAU, Julie, Anne [GB/GB]; 30 Leigh Close, Addlestone, Surrey KT15 1EL (GB).

(74) Agents: SPAARGAREN, Jerome et al.; Electronic Intellectual Property, Suite 308, The Foundry, 156 Blackfriars Road, London SE1 8EN (GB).

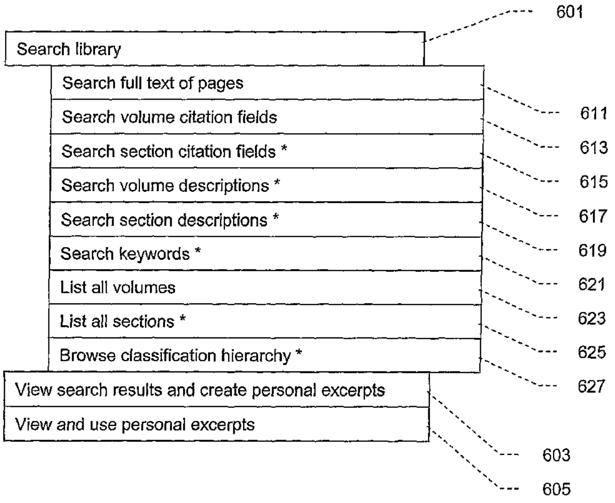
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: DIGITAL LIBRARY SYSTEM


601
Search library
Search full text of pages
Search volume citation fields
Search section citation fields *
Search volume descriptions *
Search section descriptions *
Search keywords *
List all volumes
List all sections *
Browse classification hierarchy *
View search results and create personal excerpts
View and use personal excerpts
611
613
615
617
619
621
623
625
627
603
605

(57) Abstract: An apparatus and method for setting up and operating a digital multi-media library configured in such as way as to enable the creation of custom sub-libraries. In this system users are able to create private themed sub-libraries that contain information assets that are excerpts of the main library's information assets. This is accomplished via a special proxy asset structure. The apparatus and method further enables, via use of the custom library feature and the special proxy asset structure, the deployment of digital libraries more quickly than current methods allow, and in a manner that spreads more of the set-up cost into the post-deployment period.

Digital Library System

Field of the Invention

The present invention relates to an apparatus and method for setting up and operating a digital library. More particularly, it relates to a system configured in such as way as to enable the creation of custom sub-libraries. It further relates to a method and system using custom sub-libraries to improve the cost-effectiveness of providing a digital library.

Background of the Invention

A digital library may be defined as a focused collection of digital information assets, including text, video and audio, along with computer-based processes enabling access and retrieval as well as selection, organisation, and maintenance of the collection (see Witten and Bainbridge, How to Build a Digital Library, Morgan Kaufmann Publishers, 2003).

Digital libraries can exist not only as stand-alone or networked libraries but also as components of more extensive digital information systems such as enterprise content management systems and digital publishing systems. These extended systems support additional processes related to the creation, use, version control, sharing and distribution (including sale) of information assets.

There is an increasing demand for organisations, companies and publishers to create digital libraries to hold their information assets so that they can take advantage of the benefits digital libraries bring, amongst others cost reduction, improved response times and extended geographical range of operational communities.

Furthermore, benchmarking surveys indicate that employees spend up to 40% of their time locating information they need to do their work. Digital libraries enable companies to eliminate this waste as well as to ensure the security, integrity and persistence of their information assets. By integrating digital libraries into extended digital information systems companies are able to improve the effectiveness and efficiency of information-dependent business processes by reducing their cycle time and cost and by increasing their consistency and security. The ability to share the use of such systems over wide area networks (WANs) enables companies to extend the geographical range of their operations without sacrificing process discipline, response time or information consistency. The demand for and utility of digital libraries and the systems that incorporate them or interact with them has increased in line with the development of the Internet, the increased power of computing devices, the availability of mobile computing and the falling cost of data storage.

The building of a digital library is a specialist task requiring specialist tools, methods and expertise. In practise the cost and time required to build a basic digital library generally increases linearly with the quantity of source material to be digitised. Furthermore, the versatility of the digital library is dependent on the way the data is organised and the amount of descriptive metadata that is included or catered for. The cost of creating digital libraries with complex data structures and rich metadata generally increases exponentially with the quantity of the source material to be included, as cross-references and other links internal to the data need to be maintained.

Although several commercial systems exist that support different parts of the building and deployment of digital libraries, the costs remain high enough to often put the building of a digital library beyond the means of organisations that have low income, limited reserves or a large body of material to be digitised and indexed. Alternatively, such organisations may develop libraries with reduced functionality.

The building of a digital library minimally requires the generation of digital information assets and descriptive metadata. This process is time-consuming and therefore very expensive. Typically, the process requires that physical information assets be converted into digital equivalents. For example, in the case of a digital document library deploying information assets such as books or journal volumes, the physical pages of each physical volume have to be scanned one by one using a digital scanner. In order to preserve the logical structure of the original asset, for example the articles in a journal volume, the scanning has to be performed in logical batches, and to make that possible the physical asset has to be either disassembled into logical batches or the logical breaks have to be marked up by physical means such as barcode labels. This is a labour-intensive process. In addition, data that describe each logical part have to be keyed into the digital library database so that each digital asset can be correctly identified and located in the future. If the full text is to be made searchable, then the digital page images have to be converted into electronic text, typically via the use of optical character recognition (OCR) software.

Apart from the labour cost these processes incur, every logical class of legacy asset has to be completely digitised, indexed, described and loaded before the digital library can be deployed, since a search on partial information yields results with poor utility and does not remove the requirement to search the legacy source. In consequence, digital libraries typically require a high level of investment before any operational benefit is achieved. It would be an advantage if systems could be set up in such a way that deployment timescales could be reduced. It would also be an advantage if systems could be set up and used in a way that allows some of the cost of building the digital library to be deferred to a time when the library is already providing a benefit to its users or owners (especially as these benefits may include an operational cost saving or an income opportunity).

A further problem of digital libraries is that some logical information assets can be very large data objects, for instance an electronic book can run to hundreds or thousands of pages. Handling such large objects constrains the performance of the system, e.g. it can take a long time to retrieve a large document over a network link. A user who is only interested in a small portion of the information in a large data object may still be required to retrieve the complete object, thus taxing system resources unnecessarily. It would be an advantage if the digital library could be set up in such a way that large information assets could be handled without limiting system performance or degrading the user experience.

A further problem arises when the information assets contain several different logical structures, for example, Journals might contain both articles and correspondence. These different structures require the underlying data storage to be segmented in an analogous way (e.g. by having separate database tables). Such data cannot be integrated. When the library is being built, separate processing, loading and maintenance tools must be created for each type of data with unique logical structure. Separate user interfaces are required for searching each type of logical asset. The overhead this represents in set-up cost and operational complexity often leads to compromises where the primary sections of an information source are digitised while sections of secondary importance may be discarded (e.g. journal articles are included but correspondence is not). It would be an advantage if the information assets could be represented in a way that allows all logical structures to be handled in a common way, both in system set-up and in system usage.

Given the high cost and long timescales involved in creating even a simple digital library, creating a digital library that has a complex data structure or rich metadata is rarely affordable. The low basic cost and high computational power of the infrastructure make many features possible in principle that cannot be realised in practise due to the high cost of creating the necessary base content and descriptive metadata. For example, it is possible in principle for a digital library to enable the information assets to be dynamically reorganised according to different organisational schemes, as long as the different organisational schemes have been predefined and the information assets referenced within each scheme. This could allow powerful searching, for example browsing through a hierarchy of associated keyword-based classes would be proof against changes in the terminology used in the actual textual content. However, the cost and time required to create such rich metadata is generally prohibitive, especially as the number of ways in which data can potentially be classified and organised is nearly infinite. Moreover, to be effective, such metadata has to characterise the information content at a low level of granularity. The lower this level is, the higher the investment required to create this metadata. It would be an advantage if the flexibility of digital libraries could be increased in such a way as to accommodate different user's needs for different organisational schemes while avoiding the usual penalty in cost and timescales.

Several systems and technologies have been developed in response to some of these known problems.

Many systems exist that automate aspects of the creation of digital equivalents of paper-based information assets. Scanners such as Canon's DR5020 or Kodak's 9520 scanner allow fast double-sided scanning of stacks of pages. Software products such as Adobe's Capture or ABBYY's FineReader allow the output of such scanners to be captured as single multi-page documents or a sequence of single-page documents, and enable these documents to be stored in a variety of formats (e.g. an image format such as TIF or a formatted text format such as HTML, the latter being generated via embedded OCR software). However, these systems do not eliminate the requirement to separate or mark up the source material into logical sections.

Several methods for splitting large digital objects into meaningful smaller ones are known outside of the context of digital libraries. For example, in US 2002/0184188 Mandyam *et al* disclose a method for extracting content from a document using rules that refer to code structures within the document (e.g. XML tags), and in US 6,370,553 Edwards *et al* disclose a method for creating subdocuments with active properties that enable subsequent association or reintegration of the subdocuments while component documents can be handled as documents in their own right. Such methods as these are commonly available in applications that allow editing or creation of new information assets as part of the process of building a library, preparing material for publishing or broadcasting, or creating low-level metadata for large or complex information assets. However, these methods still require some prior mark-up of the source material into logical sections.

In US 2003/0028503 Guiffrida *et al* disclose a method and system for automatically extracting metadata from electronic documents using spatial and semantic analysis. Although such techniques could be used (at least in principle) to break a data-stream into logical sections, such systems would be ineffective when the data-stream consists of assets with varying logical structure.

Software products such as Captiva's Input*Accel* or ReadSoft's Eyes & Hands enable capture of asset metadata from pre-defined areas of a scanned page. This is effective for documents such as forms that have a consistent structure, but less appropriate for variable material. These systems usually provide additional tools that allow posting of captured metadata (including the entire OCR text) directly into the repository of a digital information system (e.g. Opentext's Livelink or Documentum's Documentum 5). This posted metadata is then used as information on which to search or otherwise act, while the original linked document image file is retrieved for display. Many examples exist of systems using such metadata as indexes for scanned image files. In US 2002/0083090 Jeffrey et al disclose a system for doing this in relation to a legal contracts library, and in US 2002/0176628 Starkweather discloses a system for doing this without requiring an underlying database.

Since the effectiveness of such searches is limited by the accuracy of the metadata capture processes, it is normal for such data capture systems to provide a forms-based graphical user interface for verification of OCR accuracy, formatting, data type casting, and so forth, before the text is posted to the database. Such set-ups, though effective, require each document page to be manually verified before storage, which is very time-consuming. This methodology generally does not take account of the increasing quality of digital scanning optics and the increasing intelligence of optical character recognition software. Even if the automated processing has an accuracy of near 100%, this verification step is required before the data is posted to the repository. Systems such as Documentum 5 alleviate this problem by applying artificial intelligence (AI) methods involving semantic and syntactic analysis of the OCR text, and thereby reduce the amount of manual inspection required. Unfortunately, these high-end systems are very expensive to purchase and still require considerable effort in the configuring and training of the AI subsystem. These solutions all require a substantial investment of resources in the period before the digital assets can be made available to library users.

Several solutions have been developed to ease the problem of handling large data objects. In US 5,857,204 Kauffman et al disclose a system for breaking up large documents into smaller files of variable length to enable transfer and processing without exceeding the system's memory capacity, followed by reassembly of the document when the transfer is complete. Such methods increase the reliability of systems that handle large digital objects but they do not reduce the time taken to process or transfer a large document. In addition, they do not alleviate the system performance tax associated with handling large objects that exceed in content the information requirement of the user concerned. Several systems exist that manage large objects via Adobe's portable document format (PDF) coupled with their Acrobat Reader, a viewer for PDF documents. These systems use a content server to split up the PDF data-stream into pages (using the document's internal page-break tags), allowing the user to view one page at a time. This is a great help when viewing documents of many pages, as the user does not have to wait for the whole document to be transferred to the client workstation before the content viewing can begin. However, once the user has identified the material required, the whole document has to be downloaded as a single file (even if only a small portion is wanted), or the required portion has to be saved page-wise as a series of disjunct files (which can be tedious if the requirement is for e.g. 50 pages from a 3,000 page document).

Several inventors have noted that browsing on categories is a powerful alternative to string-searching textual content, especially where there is uncertainty about the terminology or context that applies to the information being sought. In US 6112201 Wical discloses a system that provides dynamic hierarchical browsing of a library's content. In US 5,920,864 Zhao discloses a related method. These methods require a full categorisation of the data source to be effective. The cost of defining such taxonomies and of classifying each information asset can be excessive. In addition, every time a taxonomy is updated all information assets may have to be reconsidered, which makes taxonomy maintenance very labour intensive; this problem would exist for every taxonomy applied to the information asset set. To be effective, such taxonomies have to be applied to a data source at a high resolution, further increasing the cost.

In practice, what such taxonomies achieve is to provide the user with the ability to locate a themed collection of information assets, disregarding the logical structure of the library. On this view, several inventors have considered ways of creating custom sub-libraries that are made to purpose for a specific interest group. While less immediate than using an exhaustive pre-loaded classification system, it is a less expensive approach. In US 7,778,366. Gillihan *et al* disclose a system where a librarian can create a virtual (themed) bookshelf by collating a number of information assets into a special list that can be made available to a designated group of users. In WO 00/02143 Fox *et al*, and in US 2002/0087944 David disclose methods for creating custom collections by making local copies of remote data sources and keeping them synchronised with their remote sources. In WO 02/093418 Viswanathan *et al* disclose a method for assigning a relevance rank to each item in the custom library, allowing large custom libraries to be managed. These custom library solutions suffer from a number of deficits. Generally, they have to be carefully pre-prepared by specialist librarians, rather than being created "on-the-fly" as and when needed. Furthermore, the digital assets that appear in such themed collections are still the whole logical objects of the source library. The methods for splitting documents into smaller sections as referenced earlier are designed for use by those preparing digital libraries. They are not available to the end users of a library (even a custom library), therefore from an end user's perspective the library assets have to be used in the format in which they were prepared by the provider.

There is therefore a widely recognised need for, and it would be advantageous to have, a system and method that would enable digital libraries to be built and used in a way that:

- reduces the deployment timescales, and/or
- allows some of the cost of building the digital library to be incurred in the post-deployment period, and/or
- allows handling of large information assets without degrading the user efficiency, and/or
- allows multiple kinds of logical data structures to be handled in a common way, and/or
- flexibly accommodates different users' needs for different organisational schemes without escalating the system cost

Summary of the Invention

It is an object of the invention to alleviate the problems of the prior art arrangements.

A first aspect of the present invention is an apparatus configured to operate as a digital library for enabling access to information assets, the apparatus incorporating:

a) a structuring part that provides means for representing any information asset of the library with a collection of one or more proxy assets, where the or each proxy asset consists of metadata that describes and references a data portion or an ordered plurality of data portions, where each data portion contains part of the information content of the information asset being represented; and

b) a sectioning part that provides means for creating new proxy assets such that each new proxy asset references one or several of the data portions referenced by a given proxy asset.

Preferably, the apparatus incorporates an actioning part that provides means for invoking data processing means configured to manipulate any given proxy asset or one or more data portions referenced by that proxy asset.

The information content of a library is generally regarded as being comprised of information assets, where an information asset is some piece of information that comprises a meaningful whole.

A key feature of this invention is that the information content of the library is represented by means of proxy information assets. A proxy asset does not directly contain any of the data contained within the corresponding information asset, but instead contains metadata that references an ordered plurality of data portions, where each data portion contains part of the information content of that information asset. The information contained within any one data portion need not comprise a meaningful whole, but the plurality of data portions referenced by a proxy asset, when combined in the order determined by the metadata in the proxy asset, together form a meaningful whole that corresponds to an information asset of the library. In addition, the proxy asset contains metadata identifying and optionally classifying the proxy asset.

The library may contain a proxy asset corresponding to some information asset, while also containing other proxy assets corresponding to meaningful sections of that information asset. In this case, each proxy asset corresponding to a meaningful section references, in a specific order, one or several of those data portions referenced by the proxy asset that represents the whole information asset. A section may be meaningful if it corresponds to a logical section within the information asset, or if it corresponds to an excerpt of personal interest to a user.

This representation allows a logical section within an information asset to be modelled by adding a new proxy asset rather than by changing an existing one. This is in contrast to conventional systems where an information asset is represented by a single, self-contained information unit, and a logical section within that unit is identified by means of tags or other control characters inserted amongst the information within that unit. The representation used by this invention therefore enables logical structure within a library to be refined over time without affecting existing data or existing operation of the library.

The structuring part of the invention retrieves selected information assets of the library and presents them in the structure described above. A library system designed to be an embodiment of this invention will most likely contain information stored as data portions, with appropriate metadata structured to capture the relationship between proxy assets and data portions. Alternatively, an embodiment of the invention may be integrated into an existing conventional digital library, in which case the structuring part of the invention processes conventionally stored data into the appropriate structure during retrieval.

An advantage of the invention is that data portions may reflect the modularity of the physical medium from which the information originated, rather than any inherent modularity in the information content. This allows the library provider to choose data portions that are fastest and cheapest to process into electronic form from their physical source. For example, information originating from a paper-based source could have data portions each representing the information contained in a single physical page. An embodiment of the invention may therefore be deployed much more cheaply than one in which each information asset must first be converted into a self-contained electronic form.

The structuring part of the invention may include a display part that enables a user to interact with the proxy asset metadata and any of the data portions referenced by the proxy asset. In some embodiments, a user need not be aware that the asset is a proxy one; for example with an appropriate interface a user paging through an electronic document might be unable to detect that it is a proxy document referencing a plurality of single page files rather than a true multi-page document.

It will be appreciated that, since proxy assets may reference a subset of the data portions referenced by other proxy assets, there is an implicit hierarchy between proxy assets. Proxy assets may therefore be assigned to nodes within a normal library catalogue or classification hierarchy.

The sectioning part of the invention provides means for a user to create a new proxy asset that represents an excerpt from an information asset of the library. Such a proxy asset may be a private excerpt, representing the temporary personal interests of a user, or it may become a permanent, public part of the library, representing a logical section within the information asset.

In a possible embodiment of the invention, the sectioning part may provide means for a user to create a permanent, personalised list of excerpts, similar to a reference notebook.

In another possible embodiment of the invention, the sectioning part may provide means for an administrative user to improve the library after deployment, by creating new, permanent proxy assets to capture increasingly refined logical sections within the information assets of the library. If an embodiment of the library is designed to use whatever proxy assets are available at any time, then systematic application of the means of the sectioning part will gradually increase the efficiency of the library system.

In an appropriate embodiment of the invention, the sectioning part may help end users of the library cope with any initial lack of structure in the data, as users may themselves identify logical sections within an information asset that do not yet have a corresponding proxy asset.

In an appropriate embodiment of the invention, the display and sectioning parts may provide means for a user to view and identify a portion of interest within an information asset that would be too large otherwise to manipulate conveniently.

Any embodiment of the invention may additionally contain an actioning part that enables manipulation of the data portions referenced by any given proxy asset. One example is that the data portions may be merged, in the order specified by the metadata of the proxy asset, to create a conventional, self-contained information asset.

It will be appreciated that a user who has defined a proxy asset representing an excerpt from one of the library's assets may, for example, use such actioning means to create a digital file containing that excerpt.

In an appropriate embodiment of the invention, the actioning part may provide means to help an end user cope with the fact that the proxy information assets are not self-contained files, by enabling such files to be generated.

In a possible embodiment of the invention, the actioning part may provide means to enable administrative users to generate conventional information assets from the proxy assets, to improve interoperability with or to more closely imitate the behaviour of conventional library systems.

Further aspects of the invention are set out in the appended claims, and features and advantages of the present invention will become apparent from the following description of preferred embodiments of the invention, which is given by way of example only and made with reference to the accompanying drawings.

Brief Description of the Diagrams

Figure 1 illustrates an operating environment for an embodiment of the present invention;

Figure 2 illustrates the primary components of a system that operates in accordance with an embodiment of the present invention;

Figure 3 is a flow diagram of the process for preparing the data according to this embodiment of the invention;

Figure 4 is a schema diagram illustrating an exemplary database schema supporting an embodiment of the present invention;

Figure 5 is a simplified diagram of an exemplary user interface according to this embodiment of the invention;

Figure 6 is a simplified diagram of a user interface supporting the creation of personalised sections according to this embodiment of the invention;

Figure 7 is a flow diagram illustrating full-text searching according to this embodiment of the invention;

Figure 8 shows a simplified layout for a graphical user interface for displaying and using data retrieved by a search, according to this embodiment of the invention;

Figure 9 is a flow diagram illustrating how the results list is prepared according to this embodiment of the invention.

Figure 10 is a flow diagram illustrating the creation of a personal excerpt according to this embodiment of the invention;

Figure 11 shows a simplified layout for a graphical user interface for displaying and using personalised excerpts, according to this embodiment of the invention;

Figure 12 is a flow diagram illustrating the process for saving a local copy of a personalised excerpt according to this embodiment of the invention;

Figure 13 is a simplified diagram of a graphical user interface for displaying volumes and enabling administration of data relating to a single volume, according to this embodiment of the invention;

Figure 14 is a flow diagram illustrating the creation of a public section by a method that copies an existing user excerpt according to this embodiment of the invention;

Figure 15 is a flow diagram illustrating the creation of a public section by a method that uses given section citations according to this embodiment of the invention;

Figure 16 shows a simplified layout for a graphical user interface to support the creation of a public section by a method that characterises the structure of title pages according to this embodiment of the invention;

Figure 17 is a simplified diagram of a graphical user interface for displaying sections and enabling administration of data relating to a single section, according to this embodiment of the invention.

Overview of the First Embodiment

The first embodiment of the invention is a digital document library. Such libraries are particularly valuable for providing wide access to rare, fragile or deteriorating paper-based documents, and provide for a compact alternative to the storage of bulky paper-based records.

As indicated in the background section, the conventional process for creating a digital version of a paper-based library involves a labour-intensive pre-processing phase. Physical volumes are manually separated into logical sections; for each section descriptive metadata is keyed in, the section is scanned into an image file and optionally processed by optical character recognition (OCR) into a text file.

Although the cost of this phase is high, this approach is unsurprising since many physical volumes (e.g. journals) have a well defined logical structure (e.g. articles) and there is often little significance in the physical structure of the volume (e.g. the page breaks and the chronological sequence of articles). Each logical section has well defined metadata, comprising fields such as author details, title, abstract etc. The metadata facilitate efficient searching for an individual logical section and are important for duplicating the familiar functionality of paper library catalogues and citation indices; it is therefore conventional to identify such structure and metadata as early as possible.

In contrast to known systems, this embodiment of the invention stores the information assets of the library as data portions, where each data portion holds the information contained within a single physical paper page. Each data portion is stored in two different formats thereby capturing an image of the original physical page as well as the text content of the page.

A proxy asset is created for each physical volume to be represented in the library. As every physical page is derived from a physical volume, every data portion representing a physical page is linked to at least one set of metadata characterising a proxy volume asset. Initially, no other proxy assets are created.

In a conventional digital library, when a user has searched the library's assets and identified a logical item of interest (usually a multi-page item such as an article), the digital library software may allow the user to retrieve the item. Typically, such an item is in some file format, e.g. PDF, for which it can be assumed all users will have, or be able to obtain, appropriate viewing software. The user opens the document using that secondary software, and uses that software's internal search means to find the exact locations at which the search terms appeared.

In contrast, using the first embodiment of the invention, a search identifies individual page-wise data portions satisfying the search expression. The user is presented with a list indicative of the parent proxy volume assets rather than the individual pages, but may view any of the single pages referenced by a selected proxy asset, including the specific pages identified by means of the search.

Using an appropriate interface, a user may identify a range of pages of interest within any volume, and create a new, personal proxy asset referencing that excerpt from the volume.

Over time, new proxy assets may be created to represent some of the logical sections within any volume. For example, where a physical volume is a journal volume, a new proxy asset might represent an article in that volume. Where a physical volume is a book, a new proxy asset might represent a chapter in that book, or a section within a chapter of that book. A search of the library system will return the smallest of the currently available proxy assets that reference pages that meet the search criteria. Therefore, as proxy assets are added to the library, search and browse efficiency increases.

The data portions referenced by a proxy asset may be combined into a single document which may be retrieved by a user as a local file.

Alternatively, data portions within a proxy asset may be combined and stored as additional metadata for that proxy asset. Such metadata may be full-text searched in order to search at section rather than page resolution, as is the case with conventional digital library systems.

In summary, this document library embodiment initially lacks certain features of conventional digital document libraries. These absent features diminish only the efficiency of the library, and not the core capability. Volumes are not structured into their logical sections in advance, but this is mitigated by providing the end users with the ability to create virtual sections, and structure can be added over time. Citations for sections are not initially available, but full-text searching is provided as an alternative (and arguably more powerful) method for locating items of interest, and section citations can be added over time. Full text searching requires individual pages to meet the search criteria rather than whole articles, but this is a useful feature to have and whole article searches can be added over time by merging section text into new metadata. The OCR-generated text is not necessarily proofread, but the high accuracy of modern OCR software ensures that full text searching on that text will only miss a small portion of possible hits even if simple search methods are used, and results of a conventional-library standard can be obtained by using sophisticated search methods that utilise fuzzy logic, semantic processing, specialised lexicons, etc. The ability to edit text to remove such errors ensures that search accuracy can be improved over time in libraries that use simple search engines. The raw text cannot contain images, diagrams or formatting, but this is mitigated by the supplementary availability of exact images of each original page, which are available for viewing or retrieval.

Additionally, this document library embodiment has advantages over a conventional digital document library. Any user can create a personal list of reference notes and excerpts of personal relevance. The full text for large items such as books can be made available in an efficient manner, since a user can execute a search that identifies and displays individual pages of potential interest, whereas viewing an entire large item to establish its relevance would be impractical. In addition, a large document can be viewed one page at a time and short excerpts of interest downloaded. Since all physical volumes consist of a sequence of physical pages, all can be processed in a similar manner, irrespective of the nature of their content. A single library data structure can therefore contain articles, books, correspondence, book reviews, obituaries, conference reports and so forth that can all be searched simultaneously. Moreover, such an embodiment can be implemented at a fraction of the cost of a normal digital library.

Detailed Description of the First Embodiment

The first embodiment will now be described in more detail, with reference to Figures 1 to 17.

System Overview

Figure 1 illustrates a configuration and operating environment for an embodiment of the present invention. The environment comprises a data preparation workstation 110, a scanning input device 115 that is arranged to co-operate with the workstation 110, a deployment server system 120, and a client workstation 130. Links 150 and 160 may be any form of network that supports data transfer between the systems.

To initiate data preparation, a user 170 inputs the physical pages of a physical volume to be scanned to the input device 115, whereupon a digital image file is created and stored on workstation 110. The user 170 invokes various processes on workstation 110 to process the image files, whereupon the user transfers the resultant data to the server 120 and loads at least part of the data into the database 125 on server 120, in a particular format and structure to be described later.

To operate the embodiment, a user 180 working on a client workstation 130 connects to the deployment server via the network 160. The user's actions cause client processes on workstation 130 to send requests to server processes on 120 that respond by returning data that is displayed to the user.

Figure 2 illustrates in more detail the primary components of one implementation of the environment described above. The data preparation workstation system 210 comprises a system bus 216 connecting the central processing unit (CPU), random access memory (RAM), I/O adaptors facilitating connection to user input/output devices including scanner 215, a memory adapter facilitating connection to a hard disk drive, and a network adapter facilitating interconnection with other devices on the network 250. The data preparation system's RAM 217 contains operating system software 218 which control, in a known manner, low-level operation of the workstation. The workstation includes software 211 for controlling the scanning device 215 and generating digitised images of scanned documents, image processing software 213, text processing software 214 and optical character recognition (OCR) software 212, which, as is known in the art, is for identifying text characters in an image-format computer file and generating a corresponding text-format computer file. This software is stored on the hard disk 219 and invoked via operating system processes 218 that cause them to be loaded into the computer's RAM 217. The workstation also includes client software (not shown) for remote access to the server 220.

The server 220 comprises a system bus 216 connecting the central processing unit (CPU), random access memory (RAM), a memory adapter facilitating connection to a hard disk drive, and a network adapter facilitating interconnection with other devices on both networks 250 and 260. The server RAM 227 contains operating system processes 228, database software 224, an enabling engine 223 to be described later, application server software 222 and web server software 221. The server's hard disk 229 contains the database data store 225 and various image files 226.

The client workstation system 230 comprises a system bus connecting the central processing unit (CPU), random access memory (RAM), I/O adaptors facilitating connection to user input/output devices, a memory adapter facilitating connection to a hard disk drive, and a network adapter facilitating interconnection with other devices on the network 260. The client RAM 237 contains operating system processes 238 and web browsing software 231. The structure of the network 260 is such that the web browser 231 can communicate with the web server 221 and application server 222 in the server system 220.

Setting up the Digital Library

Figure 3 is a flow diagram of the process for preparing the data according to this embodiment of the invention. The physical pages of each physical volume are scanned 301 in sequence by scanner 215 using the scanning software 211 to produce a multi-page image of each volume (e.g. in TIFF format). The multi-page image file is then processed 303 by image processing software 213 such that each page of the physical volume is represented by a separate image file (typically PDF). Each filename contains a unique identifier that indicates its sequence in the pages of the physical volume.

Simultaneously, the OCR software 212 applies optical character recognition to the image file so as to create a page-delimited text file where each delimited page corresponds to the raw text content of a single page of the original physical volume 305. The text processing software 214 then processes 307 the text into a format suitable for loading 209 into the database 225 on server 220 such that the raw text of each page is contained within a separate record in a database table, as described in more detail later. Note that this is in contrast to conventional library systems, where a database record will typically contain the text for an entire logical unit of information such as an article representing the content from multiple physical pages.

Each individual step of the foregoing process is implemented using known commercial software and methods known in the art. The text processing software is custom written for each distinct format of physical volume, but may use unsurprising methods and algorithms. It will be appreciated that this preparation phase can be automated to a significant degree, and can thus be done in less time than the data preparation required for a conventional library system.

Figure 4 illustrates an exemplary data schema suitable for this embodiment of the invention. The Volumes table 410 contains records that each hold the citation details for one physical volume that has been scanned and processed. Each record in the Pages table 420 contains the raw text 424 derived by OCR from a single physical page, as well as a reference 425 to the associated file containing the digital image of that physical page, where that file is stored on the server hard disk 226. Alternatively, the associated image file could be stored within the database as part of the record in the Pages table 420.

Each digital page record in the Pages table 420 contains a link 421 & 411 to the record in the Volumes table 410 that cites the physical volume from which that page was extracted. The digital page records from each physical volume are loaded in sequence, and the page identifier 423 indicates the position of any page in that sequence.

In a typical arrangement, all tables in Figure 40 other than the Volumes 410 and Pages 420 tables are initially empty of data, and the Section_id identifier 422 is null for all records in the Pages table.

Using and Enhancing the Digital Library

The deployment server 220 includes enabling engine 223, comprising three interacting engines: a structuring engine, a sectioning engine and an actioning engine. The structuring engine provides means to select content of interest from the library and display it in a format that is appropriate to the workings of the sectioning and actioning engines, the sectioning engine provides the means for a user to create excerpts from the selected material, and the actioning engine provides means for excerpts to be processed in various ways.

Each engine comprises a plurality of software components, each of which is a computer program performing a particular function. It will be appreciated that the invention is not limited to this specific arrangement and that each component could be made up of a plurality of programs, distributed over a plurality of networked computers.

In a preferred arrangement, the enabling engine 223 receives input data and instructions in a known manner from a user's web browser 231 via the web server 221 and application server 222. The enabling engine 223 may query the database 225 via the database software 224. The enabling engine 223 may return information to the user via the application server 222 and web server 221, using known methods. Such returned information may for example take the form of an HTML user interface dynamically generated by the application server in response to instructions from the enabling engine, and transmitted to the user's web browser by the web server.

In an alternative arrangement, the client workstation 230 may incorporate a user interface process that can communicate directly with the enabling engine 223.

Figure 5 is a simplified diagram of an exemplary top level graphical user interface for operating the enabling engine. The user selects button 501 (Use Library) in order to create and use personal excerpts. Personal excerpts are groups of pages relevant to a particular user at a particular time, and are only of interest to that user or designated others. Personal excerpts are characterised by means of metadata stored in table 440.

The user selects button 503 (Administrate Library Database) in order to create or enhance permanent, public sections. Public sections reflect an inherent logical structure within a volume, for example an article in a journal volume or a chapter in a book, and once created are permanently accessible to all users. The public sections are characterised by means of metadata stored in tables 430, 450, 460, 470 and 480 of Figure 4. Some of the features of the enabling engine are dependent on whether or not corresponding data are available in these tables. In the diagrams, such data-dependent features are indicated with an asterisk.

Using the Digital Library

Upon selecting button 501, the user is presented with a menu of options as illustrated in Figure 6. Choosing option 601 enables the means of the structuring engine to search for relevant material while option 603 enables the means to display that material to the user and allow the user to create excerpts by means of the sectioning engine. Option 605 enables the means of the structuring engine to display the user's excerpts and facilitate their use by means of the actioning engine. Each option marked with an asterisk is only made available to the user if there is at least one record in its corresponding data table (the correspondences between options and data tables will be described later). Options without asterisks are always available given the initial data configuration described above.

Search Library

Selecting the Search option 601 results in a submenu of options as shown in Figure 6. Figure 7 is a flow diagram illustrating the action of the structuring engine after the user has selected menu option 611 (Search full text of pages). Process block 701 involves presenting an interface to the user (not shown) whereby the user can type in or assemble a Boolean string expression characterising the desired search. By default, the search expression is matched against each page record in the database representing to each physical page from the physical volume. The structuring engine assembles 703 a query to send to the database 225, to instruct it to search through the Page_text field 424 of the Pages table 420, for all individual pages that contain text that matches the given expression, and to return to the engine the Page_id 423, the Volume_id 421 and the Section_id 422 of matching pages.

When initially deployed, the Section_id field is null for all pages in the Page table. However, as will be described later, the sectioning engine, or some other method, may be used to create public sections representing logical groups of pages. If a page has been incorporated into such a section, the page's Section_id 422 will reference a record in the Sections table 430 via field 431, said record capturing the citation information for that section. Each page is therefore uniquely associated with a volume, and may also be uniquely associated with a section, which is itself uniquely associated with that same volume. Some of the pages matching the search expression may be associated with the same volume, and possibly also the same section.

The structuring engine separates the matching pages into two groups depending on whether the Section_id is null or non-null. For the former group, the structuring engine compiles 705 a list of the distinct identifiers of all volumes containing at least one matching page. For the latter, the structuring engine compiles 707 a list of the distinct identifiers of all sections containing at least one matching page. Each identifier, whether section or volume, is associated 709 with a collection of Page_ids representing all of the pages within the given section or volume that match the search expression.

The above database search can be limited with additional constraints in the usual manner, e.g. by constraining the values of citation fields in the Volume table such as publication year, or constraining the search to volumes identified in the previous search, or to pages referenced in the User Excerpts table 440, which is described later.

The database schema in Figure 4 contains a Merged Section Text table 480 that is initially empty. Authorised users may have used the actioning engine in a manner to be described later, or some other means, to populate this table with data. Each record contains, in the Section_text field 482, the full text of an entire section. If such data exists, the user interface contains a control allowing the user to specify that the structuring engine should apply the search criterion to a whole section rather than to individual pages. This corresponds to the conventional way of full text searching an electronic library.

In this case, process block 711 is activated, whereby the structuring engine assembles a query to send to the database 225, to instruct it to search through the Section_text field 482 of the Merged Section Text table 480, for all sections that contain text that matches the given expression, and to return to the engine the Section_id 481 of the matching sections. It will be appreciated that Page_ids cannot be identified under these circumstances.

Returning to the search submenu in Figure 6, if the user selects option 613 (Search volume citation fields), the structuring engine provides an interface for capturing the user's volume search expression, which will contain a combination of requirements for the various citation fields in the Volumes table 410. The structuring engine queries the database to identify Volumes table records with fields matching the given expression.

If there is data in the Section table, the user may select option 615 (Search section citation fields). In this case, the structuring engine provides an interface for capturing the user's section search expression and identifies those records in the Sections table 430 with fields matching the given expression.

The database schema in Figure 4 contains a Volume Description table 450 and a Section Description table 460, both of which are initially empty. Authorised users may have used the sectioning engine in a manner to be described later, or some other means, to populate these tables with data such as abstracts for journal articles or reviews of books.

If there is data in the Volume Descriptions table, the user may select search option 617 (Search volume descriptions), in which case the structuring engine instructs the database to full-text search the Volume_description field 452 using a search string captured from the user. The process results in a list of volume identifiers indicating matching volumes.

If there is data in the Section Descriptions table, the user may select search option 619 (Search section descriptions), in which case the structuring engine instructs the database to full-text search the Section_description field 472 using a search string captured from the user. The process results in a list of section identifiers indicating matching sections.

The Keywords table 460 has many-to-many relationships to the Volumes and Sections tables, and contains keywords associated with volumes and/or sections. If there are keywords in this table, the user may select option 621 (Search keywords). The structuring engine captures a keyword from the user, queries the database to identify volumes and sections linked to that keyword, and assembles a collection of volume and section identifiers as before. There are many ways of modelling classification hierarchies in digital libraries. Any such hierarchy can be linked into the core components of this embodiment of the invention. For example, the Keywords table 460 can function as a classification hierarchy, since keywords within the table may be linked to parent keywords and keyword aliases within the same table. Option 627 triggers the structuring engine to produce a traversable tree view of keywords and their aliases by methods well known in the art, allowing volumes or sections to be identified according to their allocation in one or more classification systems.

Options 623 (List all volumes) and 625 (List all sections) trigger the structuring engine to assemble identifiers for all volumes or all sections respectively.

Various security techniques not part of this invention may be used to ensure that the engine only accesses documents that the user has permission to access.

Display Search Results

Once the structuring engine has identified a collection of section and/or volume identifiers, each one possibly having an associated collection of page identifiers, the user is returned to the menu of Figure 6. Selecting option 603 instructs the engine to present that material to the user and to allow the user to create personal excerpts.

Figure 8 is a simplified conceptual diagram of a graphical user interface for displaying and using the retrieved data. Area 801 is used to display a list of the volumes or sections containing matching pages, while area 803 is used to display a single page.

Figure 9 is a flow diagram illustrating the action of the structuring engine in populating the display area 801. By process block 901, for all identified Section_ids 431, the structuring engine lists in area 801 the corresponding section's title 433 concatenated with the title 412 of the volume to which that section belongs, said volume title identified through the linking fields Volume_id 432 and 411. Other citation details from the Section and Volume tables may be displayed with each section title.

Next, by process block 903, for all identified Volume_ids, the structuring engine appends to the list 801 each volume title 412 together with any additional desired citation information for that volume.

Each list item 805 may be associated with a collection of Page_ids 423 indicating the pages in that volume (or that section if the list item is a section) that match the search expression. The user may select any one of the list items by some means such as an adjacent button or hyperlink. If the selected item has an associated collection of page_ids, the Page_text 424 of the page with the lowest of those page_ids is displayed in the area 803. If the item does not have associated page_ids, the text of the first page record in that volume (or section) is displayed.

Button 807 allows the user to switch between viewing the raw, unformatted text from the Page_text field 424, or viewing the image of that page. The page image has the advantage of being an exact reproduction of the original physical page of the physical volume. The raw page text is unformatted, may contain scanning inaccuracies, and cannot accurately reproduce any photographs, diagrams or tables that may be embedded in the source page. However, the structuring engine can highlight search terms in the raw text, and the user might be allowed to copy sections of text to the computer's clipboard for use in compiling research notes. It is therefore useful for the user to be able to choose between these two views for any page. If the button 807 is selected, the structuring engine retrieves from the database the Page_image_path 425, which specifies the path and filename to the image file for that particular page. This image file is then retrieved from that location and displayed in the area 803.

The two buttons 813 instruct the structuring engine to display the page preceding or following the current page, from the sequence of pages of the selected volume or section. In each case, a new page_id is calculated by incrementing or decrementing the current page's page_id, and the corresponding page text or image is retrieved and displayed.

The two buttons 811 instruct the structuring engine to display, if available, the previous or next page out of the list of those pages that matched the search expression and were within the selected volume or section.

It will be appreciated that since only one page at a time is retrieved and presented to the user, it is possible to deploy large volumes such as books in this manner, without the user having to retrieve the entire volume before being able to read any part of it.

Create Personal Excerpts

Buttons 821, 823 and 825 allow a user to create a personal excerpt from the selected volume, according to the process illustrated in Figure 10. The user uses buttons 811 and 813 to navigate to the first page of the desired excerpt, and then presses button 821. The sectioning engine stores 1001 the Page_id in memory. Then the user navigates to the last page of the desired excerpt and presses button 823, whereupon this Page_id is also stored 1003 in memory. If the user then presses button 825, the user is offered 1005 an interface (not shown) for typing in personal metadata for the excerpt, e.g. a title. The sectioning engine instructs 1007 the database to insert a new record in the User Excerpts table 440 containing the user's User ID 441, the volume ID 442 and any section ID 443, the excerpt's first page ID 444 and last page ID 445, along with additional metadata fields, e.g. the excerpt title in field 446.

Alternatively, with appropriate modifications to the User Excerpts table, a single user excerpt could reference multiple distinct page ranges within a volume.

It will be appreciated that the user except could be simply a group of pages containing information that is temporarily of interest to the user. It could also correspond to a logical group of pages, e.g. an article in a journal volume, where that logical group has not yet been captured as a public section in the Section table 430. It will be appreciated that, in this way, the facility for the user to make personal excerpts mitigates against any initial lack of structure in the way volumes are stored.

Display and Use Personal Excerpts

Selecting menu option 605 instructs the structuring engine to display all of a user's personal excerpts. Figure 11 is a simplified conceptual diagram of a graphical user interface for displaying and using personalised excerpts. Area 1101 is used to display a list of volume titles, or section plus volume titles, derived from data in table 440 processed as described above with reference to Figure 9. Each list item additionally displays the excerpt page range and metadata such as the personal excerpt title 446. A page of a selected excerpt is displayed in area 1103, and the user may page forwards and backwards within the excerpt, and change the view format of any page, as before. The user may also create an excerpt of the excerpt, by the method described above. Pressing button 1105 invokes a dialogue box (not shown) where the user may view and edit the metadata describing this excerpt.

The user may press button 1107 to save the contents of a selected excerpt as a local computer file, by a process illustrated in Figure 12. The actioning engine retrieves 1201 the volume_id 442, start_page_id 444 and end_page_id 445 from the User Excerpts table 440. The user specifies 1203 whether the excerpt is required as raw text or as an image document. If raw text is required, the actioning engine retrieves 1205 the Page_text 424 from the Page table for each page with the given Volume_id and with Page_id between the excerpt's start and end page IDs. The pages of text are joined in sequence 1207 into a single document, and an external module not part of this invention is invoked to stream 1209 the document to the user's computer. If an image document is required, the actioning engine retrieves 1215 the path and filename 425 for the image files of each page in the excerpt and invokes external modules not part of this invention to merge in sequence 1217 the separate image files into a single multi-page image file and stream 1219 it to the user.

Various alternative processing options may be applied to an excerpt by means of the actioning engine, for example statistics such as word count or word distribution can be computed, the excerpt can be passed to an external module for translation into another language, or an internet search can be triggered using high-prominence terms detected in the extract.

Administrating the Digital Library

Returning to the main user interface in Figure 5, users with appropriate authorisation may select the option 503 to administrate the database supporting this embodiment of the invention, and in particular to create public sections. Upon selecting 503, the user is presented with a user interface represented by the simplified conceptual diagram in Figure 13. Area 1301 is populated with a list of the titles of all volumes in the database, extracted from the Volume table 410 by means of the structuring engine. The user can select a particular volume from the list, whereupon the illustrated buttons become selectable.

Create New Public Section

If the user presses button 1302 (Create new public section), the sectioning engine allows the user to choose one of the available methods for creating a new section according to the invention. Three alternative methods are described below, by way of example.

The user may invoke a method that copies an existing personal excerpt, as illustrated by Figure 14. In this case, the sectioning engine presents 1401 to the user a text box (not shown) in which the user can type in an identification to indicate another user whose personal excerpt is to be copied. The sectioning engine lists 1403 all excerpts defined for the selected volume by the given user, as defined in table 440. If the user selects an excerpt and confirms the section creation, the sectioning engine instructs 1405 the database to insert a record into the Section table 430 under a new Section_id 431, said record containing the current volume's Volume_id and the personal excerpt's title 446 as the Section_title 433. The sectioning engine then instructs 1407 the database to update the Section_id field 422 in the Pages table 420 so that it equals the newly created Section_id for all pages with Page_ids between the personal excerpt's start 444 and end 445 pages.

Alternatively, the user may invoke a method that uses given section citations, illustrated by Figure 15. The sectioning engine reads 1501 a section title, author and number of pages from a previously created file or database table. It instructs 1503 the database to search the Page_text field 424 of records in the Pages table 420 belonging to the selected volume, that have not already been assigned to a section, for pages containing the author's name and title text. Information about the format of the title page may be used to ensure a unique page is identified 1505. The sectioning engine stores the page identifier and calculates 1507 the last page of the section using the given number of pages. The sectioning engine instructs 1509 the database to insert a new record containing the section title into Section table 430, with a new Section_id 431. The sectioning engine then instructs 1511 the database to update the Section_id field 422 in the Pages table 420 to equal the newly created Section_id for all pages between the section's start and end pages.

Alternatively, the user may invoke a method that recognises title pages. Figure 16 is a simplified diagram of a graphical user interface to support this method. If the user presses button 1601, the sectioning engine presents an interface (not shown) into which the user types a Boolean expression representing a text pattern that is known to appear on all section title pages in that volume, e.g. the string 'pp.'. The sectioning engine instructs the database to search the Page_text field 424 of records in the Page table 420 belonging to the selected volume, that have not already been assigned to a section. The sectioning engine stores the identifiers of matching pages and displays the first such page in area 1611. The user may navigate through these title pages using the buttons 1602. For a displayed title page, the user may press button 1604, whereupon the sectioning engine provides an interface for the user to type in citation metadata for the section beginning with that page. Alternatively, the sectioning engine can call an external module not part of this invention to extract the title, author and number of pages in the section from the title page text. Such modules are known in the field of string manipulation and are the focus of ongoing development. If the user then presses button 1605, the sectioning engine invokes a method to find and display the last page of the section starting with that title page. This may involve using the number of pages in the section, as extracted from the title page, to calculate the end page, or simply moving to the page prior to the next section title page. The user confirms the section details by pressing button 1606, whereupon the section title and other citation metadata is inserted into the Section table 430 as a record with a new Section_id 431. The sectioning engine then updates the Section_id field 422 in the Pages table 420 so that it equals the newly created Section_id for all pages between the section's start and end pages.

Other Volume Functions

With a volume selected, the user may also press button 1305 to add or edit volume citation metadata. The actioning engine captures such information and instructs the database to update the appropriate records in the Volume table 410. The user may press button 1306 to add a new record to the Volume Descriptions table 450, or to edit an existing record. Button 1304 invokes a process that allows the user to assign keywords from the Keywords table 460 to this volume, or to add new keywords to the Keywords table. Button 1307 invokes a user interface allowing the user to search or browse through individual pages within the volume, where each page is displayed in an editable window. The user may make changes to the page text, for example to correct OCR errors or to enhance the text display by adding HTML formatting, and instruct the actioning engine to update the corresponding record in the Pages table 420 accordingly.

Section Functions

Pressing button 1303 instructs the structuring engine to list all public sections defined within the selected volume. Figure 17 is a simplified diagram of a graphical user interface to support the administration of public sections. The sections defined for the selected volume are listed in area 1701.

With a section selected, the user may press button 1706 to add or edit section citation metadata. The actioning engine captures such information and instructs the database to update the appropriate record in the Section table 430. The user may press button 1707 to add a new record containing a section abstract or review to the Section Descriptions table 470, or to edit an existing record. Button 1705 invokes a method allow the user to assign keywords from the Keywords table 460 to this section, or to add new keywords to the Keywords table.

Button 1703 (Create merged section text) causes the actioning engine to invoke a process to create a single searchable record containing the text of the entire section. Firstly, the sectioning engine retrieves the Page_text 424 for all pages in the selected section. It then concatenates the pages in sequence to form a single string. This is inserted as a new record into the Merged Section Text table 480 and linked to the Section table via the section_ids 481 & 431. Button 1704 invokes an interface (not shown) through which the user can edit the merged section text, whereupon the record in table 480 is updated.

Second Embodiment

A second embodiment will now be described, which is generally similar to the first embodiment, for which like parts have been given like reference numerals and will not be described in further detail. The second embodiment applies to a digital document library deploying documents that are already available in electronic form but where the internal logical structure of the documents has not been identified.

In this embodiment, the structuring engine splits each document programmatically into data portions. If the content is unstructured but the file is in a multi-page format, it is split into separate page-sized files using known methods. If the content and the file are both unstructured, it is split into approximate page-sized files by splitting the file at every first blank line after a suitably-sized batch of lines. If the content has some programmatically recognisable structure, e.g. an encyclopaedia, dictionary, recipe book etc, it is split such that each structural part corresponds to one data portion.

It will be recognised that the latter data portions may not be of equal size, and the size may not approximate a paper page or display page. The structuring engine displays each data portion as if it were a page (it being appreciated that a page may be larger than the display panel in the viewer, and scrollbar or zoom features can be used to enable the user to view all of the page information). Alternatively, the structuring engine includes a page server that dynamically splits these data portions into display-sized pages using known methods. These served pages may be considered virtual data portions.

Information stored in this form may be searched, displayed, sectioned, listed and processed as described above with reference to Figures 1 to 17.

Each original electronic document may be replaced by the corresponding data portions, or, if derived from an existing conventional digital library, it may be retained in that library. By employing the structuring engine to display the document in a form compatible with the invention, the additional features of the invention may be added to the conventional document library.

Third Embodiment

A third embodiment will now be described, which is generally similar to the first embodiment, for which like parts have been given like reference numerals and will not be described in further detail. The third embodiment involves a more sophisticated distribution of data and engines between the hardware components of the system.

In this embodiment, the client workstation 230 includes a version of the enabling engine arranged to communicate with a local database. The workstation also includes a user interface program arranged to communicate with the remote sectioning engine 223 as well as the local sectioning engine.

The user interface can interact with the remote enabling engine, which in turn interacts with the remote database, in the manner of the first embodiment. In addition, the user interface can interact in the same way with the local enabling engine, which interacts with the local database. The user interface can cause the two enabling engines to synchronise the user's personal excerpt lists between the two databases, using known methods. The volume and section metadata and the page data and files referred to by the personal excerpts in the list are synchronised, therefore both databases contain copies of all data relating to the excerpts most recently defined by that user (or including other users if authorised). Whenever the remote database is unavailable, for example when the network link 260 is broken, the local enabling engine may still be used to search, view, excerpt and process the material that is in the local database.

In an alternative arrangement, the remote enabling engine and database may be replaced by a remote enabling application programmers interface (API) to an alternative type of digital library not implemented according to this invention. The API allows the local enabling engine to search, display, section and process data from the alternative library, in the manner of the first embodiment. Data may be extracted from the remote library system and saved in the local database in a manner consistent with the first embodiment.

In an alternative arrangement, the local enabling engine is arranged to copy and save data relating to user extracts from multiple remote enabling engines and from multiple remote sectioning APIs. It will be appreciated that, in this case, the system comprising local interface, database and enabling engine becomes a centralised, interactive store for a user's personal excerpts taken from a multiplicity of different remote digital libraries.

Fourth Embodiment

A fourth embodiment will now be described, which is generally similar to the first embodiment, for which like parts have been given like reference numerals and will not be described in further detail.

The fourth embodiment is an internet publishing centre enabling cartoon artists to self-publish their material in a collective, themed environment. In this embodiment, a data portion corresponds to a single cartoon strip, while each initial proxy asset references all of an artist's cartoons for one year, in chronological order. Users create additional proxy assets representing, for example, cartoons on a common theme, or strips that develop a running story.

In this embodiment, an enabling engine is running on a centralised server 220. Various artists each have data preparation systems similar to 210, at which they scan cartoon strips as they finish drawing them. The strips may have varying length, may be in colour or black and white, and may have any layout. Each strip is saved as a separate image file, as described in the case of the first embodiment.

The artist may optionally OCR the cartoons to extract and store the strip text from the speech bubbles; this is a feasible technique as some OCR packages can be trained to recognise a consistent handwriting.

The server 220 runs a loading engine which interacts with the application server and web server to dynamically generate a user interface that is presented to the artist's web browser for display, where the web browser is running on the artist's data preparation system 210.

Through the loading engine interface, the artist can copy the cartoon strip image files onto the server's file system, and insert a reference to each image file as a separate strip record into a table analogous to the page table 420 in the database on the server 220. The loading interface enables the artist to additionally include the strip text that was generated by the OCR process into the corresponding strip record. Alternatively the artist may manually enter the cartoon strip's text through the user interface. Alternatively, a centralised OCR process on 220 may be invoked by the loading engine to generate strip text automatically upon loading of a new image file, and to store that strip text in the appropriate strip record in the page table.

The strip records for each artist are loaded chronologically and sequenced in that order. As each strip record is loaded, it is linked to an annual collection record in a table analogous to the volume table 410. The strip collection record contains metadata recording the artist and year, and any other descriptive information relevant to that year's collection of cartoon strips by that artist. Note that the strip records for the various artists are stored in the same data table.

The artist or an administrator may use the means provided by the sectioning engine to create a logical section that is a subset of an annual collection, e.g. a monthly collection.

The artist additionally uses the means provided by the sectioning engine to create themed sequences of one of more cartoon strips. These may correspond to a sequence of strips telling an extended story, or to cartoons on a common subject, and may include metadata describing the theme. It will be appreciated that such themed sequences, although ordered, need not be sub-sequences of chronological collections, and may include cartoons in any specified order extracted from various annual collections. The artist's themed sequences are designated as accessible to all users of the cartoon publishing centre.

A user of the centre is a person who subscribes to the service provided by the centre, namely to be able to browse or search the database to see cartoons drawn by any or all of the participating artists. The user accesses the embodiment from a workstation similar to 230, by means of a user interface generated by means of the enabling engine 223 and application server 222, and displayed in the user's web browser. Users can browse the themed sections created by the artists, and view the image files to see the cartoons containing that text. Users can additionally search the strip text for cartoons with text containing keywords of interest.

Artists may use the means of the enabling engine to edit a common classification hierarchy and to add theme references to the nodes of this hierarchy.

Users may navigate this hierarchy to see themed sequences from various artists on similar themes conveniently grouped together.

In addition, users may create personal themed collections of cartoons for future reference or to download or to share with authorised friends. These may include cartoons from any of the participating artists.

It is to be understood that any feature described in relation to any one embodiment may be used alòne, or in combination with other features described, and may also be used in combination with one or more features of any other of the embodiments, or any combination of any other of the embodiments. . Although the preferred embodiments of the present invention have been described and illustrated in detail, it will be evident to those skilled in the art that various modifications and changes may be made thereto without departing from the spirit and scope of the invention as set forth in the appended claims and equivalents thereof.

Claims

1. An apparatus configured to operate as a digital library system for enabling access to information assets, the apparatus incorporating:

    (a) a structuring part that provides means for representing an information asset of the library with a collection of one or more proxy assets, where the or each proxy asset consists of metadata that describes and references a data portion or an ordered plurality of data portions, where each data portion contains part of the information content of the information asset being represented; and

    (b) a sectioning part that provides means for creating new proxy assets such that each new proxy asset references one or several of the data portions referenced by a given proxy asset.

2. The apparatus of claim 1 wherein the structuring part includes means to extract the desired content from one or several local or distributed data repositories.

3. The apparatus of claim 2 wherein some or all of the data repositories are exclusive to the digital library system.

4. The apparatus of claim 2 wherein some or all of the data repositories are shared with one or several other networked digital information systems.

5. The apparatus of claim 4 wherein the digital information systems in the network of linked systems have disjunct sets of information

6. The apparatus of claim 4 wherein the digital library system's information assets duplicate some or all of the information from other information systems to which they may be permanently, temporarily or sporadically connected.

7. The apparatus of any of claims 2 to 6 wherein some or all the data repositories are permanently accessible to the digital library system.

8. The apparatus of any of claims 2 to 6 wherein some or all of the data repositories are only temporarily or sporadically accessible to the digital library system.

9. The apparatus of any of claims 2 to 8 wherein some or all of the data repositories contain information assets structured in a way that mirrors the desired representation.

10. The apparatus of any of claims 2 to 8 wherein some or all of the data repositories contain one or several information assets structured in a way that does not mirror the desired representation and where the structuring part incorporates means to generate one or several proxy assets that represent the information content of these information assets in the desired manner.

11. The apparatus of claim 10 wherein the proxy asset generating means incorporates means to segment the or each information asset into data portions using code parts internal to the asset.

12. The apparatus of claim 10 or 11 wherein the proxy asset generating means incorporates means to segment the or each information asset into data portions using data parsing techniques configured to recognise ordinary formatting or structural features internal to the asset.

13. The apparatus of claim 10 wherein the proxy asset generating means incorporates means to segment the or each information asset into data portions each containing the same number of characters.

14. The apparatus of claim 10 wherein the proxy asset generating means incorporates means to segment the or each information asset into data portions appropriately sized to suit the user interface display.

15. The apparatus of any of claims 10 to 14 incorporating further means whereby the or each proxy asset and referenced data portions, once generated, can be stored in a structure that preserves that desired representation.

16. The apparatus of any preceding claim wherein information content represented by any data portion may be available to the structuring part in one or several formats, and the structuring part includes means whereby any of the available formats can be used in the representation of that data portion.

17. The apparatus of any preceding claim wherein different information assets are owned and managed by different organisations and/or individuals.

18. The apparatus of claim 17 incorporating means whereby the or each referenced organisation or individual may control user access to their information assets.

19. The apparatus of any preceding claim wherein the data repository or repositories may exclusively contain data relating to a specific theme.

20. The apparatus of any preceding claim wherein the structuring part incorporates means to select some or all of the proxy assets and further means to display a list of references to the selected proxy assets.

21. The apparatus of claim 20 wherein the selected one or several proxy assets represent the result of a search invoked either by a user or by an automated process and wherein the search is constrained to a specified subset of the proxy assets.

22. The apparatus of claim 21 incorporating means to apply the search to the data portions referenced by the specified subset of proxy assets and incorporating further means to assemble and display a list of references to those proxy assets that reference data portions that complied with the search criteria.

23. The apparatus of claim 22 wherein the data portions reference text data and the search criterion is an expression incorporating string functions that characterises compliant units.

24. The apparatus of claim 22 wherein the data portions reference multimedia data of any kind and the search criterion is an expression incorporating functions that characterises compliant units.

25. The apparatus of any of claims 22 to 24 wherein the list assembly means incorporates means to identify for each compliant data portion the smallest proxy asset referencing that data portion, where the smallest proxy asset is the one that references the fewest data portions, and further means to assemble a list containing references to the identified smallest proxy assets.

26. The apparatus of any of claims 22 to 24 wherein the list assembly means incorporates means to identify for each compliant data portion all proxy assets referencing that data portion and further means to assemble a hierarchical list containing references to those proxy assets.

27. The apparatus of any of claims 22 to 26 wherein the or each proxy asset referenced in the list is temporarily associated with further references that indicate which of the data portions referenced by that proxy asset complied with the search criteria.

28. The apparatus of claim 27 incorporating means to store in a repository of the digital library system a persistent record identifying the data portions that complied with the search criteria together with information identifying the user who or program

that invoked the search together with information identifying the search criteria specified by the user or invoking program.

29. The apparatus of claim 21 incorporating means to apply the search to the metadata component of the assets, and incorporating further means to assemble and display a list of references to compliant assets.

30. The apparatus of claim 29 incorporating means whereby the search engine additionally extends the search by automatically generating new search criteria derived from the original search criterion by the application of rules that may reference one or several internal or external lexicons.

31. The apparatus of any of claims 21 to 30 incorporating means to invoke a plurality of search engines so that each search engine searches information assets stored in a different data format.

32. The apparatus of any of claims 21 to 30 incorporating means to constrain the search to those information assets designated accessible to the user or process.

33. The apparatus of claim 20 wherein the selection is a set of proxy assets created by means of the sectioning part.

34. The apparatus of claim 33 incorporating means to constrain the selection to the set of proxy assets created by actions of the present operator.

35. The apparatus of claim 33 incorporating means to constrain the selection to the set of proxy assets designated accessible to the present operator.

36. The apparatus of claim 20 wherein the selection is the set of proxy assets that each reference a subset of the data portions referenced by the given proxy asset, excluding any proxy assets that reference a subset of the data portions referenced by any other members of that set of proxy assets.

37. The apparatus of claim 20 wherein the selection is a collection of proxy assets previously associated with a given node in a classification hierarchy.

38. The apparatus of any preceding claim wherein the structuring part incorporates means to display any of the data portions referenced by a selected proxy asset in any of the formats in which that data portion is available.

39. The apparatus of claim 38 incorporating means to identify those data portions that previously complied with a search criterion.

40. The apparatus of claim 38 or 39 incorporating further means to display the text information present in any data portion in any one of a selection of human languages.

41. The apparatus of any previous claim wherein a new proxy asset created by means of the sectioning part represents a logical section that exists within the information content represented by the given proxy asset, and the metadata for the new proxy asset (known herein as a logical proxy asset) may include a citation for and/or a description of that logical section.

42. The apparatus of claim 41 incorporating further means to progressively refine the logical structure of the digital library by enabling the systematic iterative creation of logical proxy assets and the storage of information characterising these new proxy assets in a repository of the library system.

43. The apparatus of claim 42 wherein the further means incorporates a computer program configured to create new logical proxy assets that correspond to logical sections identified within a given proxy asset by the application of encoded rules determined and parameterised by an administrative user, and to apply such processing iteratively to each of the proxy assets in a user-selected batch of proxy assets.

44. The apparatus of claim 42 where the further means incorporates a computer program and user interface configured to support users in the systematic iterative identification of logical sections within proxy assets and the subsequent creation of new logical proxy assets that correspond to the identified logical section.

45. The apparatus of any of claims 1 to 40 wherein a new proxy asset created by means of the sectioning part represents information content of relevance to a specific user or user-group, and the metadata for the new proxy asset (known herein as a themed proxy asset) identifies the creating user and may in addition include information provided by that user to characterise an aspect of the information content.

46. The apparatus of claim 45 incorporating a computer program configured to provide means to identify a meaningful sequence of data portions surrounding each data portion that conforms to the interests declared by a user.

47. The apparatus of claim 45 incorporating a computer program and user interface configured to provide means to support a user in the identification of sequences of data portions that conform to their respective interests.

48. The apparatus of any of claims 45 to 47 incorporating further means for storing information characterising an aspect of the themed proxy assets in a repository of the library system.

49. The apparatus of any previous claim wherein a new proxy asset created by means of the sectioning part references an ordered plurality of data portions that form an incremental sequence and the reference consists of references to the first and last data portions in the sequence.

50. The apparatus of any of claims 1 to 48 wherein a new proxy asset created by means of the sectioning part references data portions by individually referencing every relevant data portion.

51. The apparatus of claim 49 or 50 wherein the references to the data portions are indexes to the position of those data portions within the ordered plurality of data portions referenced by the given proxy asset.

52. The apparatus of claim 49, 50 or 51 incorporating means to establish references to data portions via an incorporated user interface through which a user can identify the data portions to be referenced.

53. The apparatus of claim 49, 50 or 51 incorporating means to establish references to data portions via an incorporated computer programme coded to identify, on the basis of predetermined rules parameterised by a user, the data portions to be referenced.

54. The apparatus of any previous claim further incorporating an actioning part that provides means for invoking data processing means configured to manipulate any given proxy asset or one or more data portions referenced by that proxy asset.

55. The apparatus of claim 54 wherein the data processing means is configured to sequentially join the data portions referenced by the proxy asset into a new temporary data portion.

56. The apparatus of claim 55 wherein the data processing means is configured to transmit the information content represented by the new data portion as a data stream in a format specified by the user to a location specified by the user.

57. The apparatus of claim 56 wherein the data stream format corresponds to one of the formats in which the data portion content is stored within the one or several repositories of the digital library system.

58. The apparatus of claim 56 wherein the specified data stream format does not correspond to any of the formats in which the data portion content is stored within the or any of the repositories of the digital library and the data processing means is additionally configured to translate the data stream into the required format before transmission.

59. The apparatus of claim 56, 57 or 58 wherein the data processing means is additionally configured to translate text information present in the proxy asset into a selected human language before transmission.

60. The apparatus of any of claims 56 to 59 incorporating further means for the user to save the data stream as a file with a name determined by the user.

61. The apparatus of any of claims 55 to 60 incorporating further means to create additional metadata for the given proxy asset by storing in a repository of the library system the text present in the temporary data portion as additional metadata of the proxy asset.

62. The apparatus of claim 61 incorporating further means to progressively improve the versatility of the digital library's search means by enabling the systematic iterative creation of textual metadata corresponding to the combined text information referenced by each of the proxy assets in a selected batch of proxy assets and the storage of this metadata in a repository of the library system.

63. The apparatus of claim 62 wherein the further means is a computer program configured to iteratively create such metadata for each of the proxy assets in a user-selected batch of proxy assets.

64. The apparatus of claim 62 wherein the further means is a computer program and user interface configured to support users in the systematic iteration of a method that creates such metadata for proxy assets.

65. The apparatus of any of claims 55 to 64 further incorporating means to store the temporary data portion as a persistent data portion in a repository of the digital library together with metadata that associates the new data portion with the given proxy asset.

66. The apparatus of any of claims 54 to 65 wherein the data processing means is configured to enable alteration of any data portion selected from those referenced by the given proxy asset.

67. The apparatus of claim 66 wherein the alteration means enables quality-enhancing editing of the information content represented by any data portion.

68. The apparatus of claim 67 incorporating further means to progressively improve the quality of the digital library system by enabling systematic iterative quality-enhancing alteration of the stored content of data portions.

69. The apparatus of claim 68 wherein the further means is a computer program configured to iteratively improve the quality of the stored content of each of the data portions in a user-selected batch of data portions.

70. The apparatus of claim 68 wherein the further means is a computer program and user interface configured to support users in the systematic iteration of a method that improves the quality of the stored content of data portions.

71. The apparatus of any of claims 66 to 68 wherein the alteration means is configured to enable editing that improves the readability of the information content referenced by any data portion.

72. The apparatus of claim 71 incorporating further means to progressively improve the readability of the information content of one or a series of proxy assets by enabling

the systematic iterative alteration the stored content of any of the data portions referenced by the or each of the proxy assets.

73. The apparatus of claim 72 wherein the further means is a computer program configured to iteratively improve the readability of the information content of each of the proxy assets in a batch of proxy assets selected by the user.

74. The apparatus of claim 72 wherein the further means is a computer program and user interface configured to support users in the systematic iteration of a method that improves the readability of the information content of proxy assets.

75. The apparatus of any of claims 66 to 74 wherein the alteration means is configured to enable the replication, in an additional format, of the information content referenced by the given data portion.

76. The apparatus of claim 75 incorporating further means to progressively improve the performance of the digital library system by enabling systematic iterative replication of the stored content in more efficient data formats.

77. The apparatus of claim 76 wherein the further means is a computer program configured to iteratively replicate the stored content of each of the proxy assets in a user-selected batch of proxy assets into a more efficient data format selected by the user.

78. The apparatus of claim 76 wherein the further means is a computer program and user interface configured to support users in the systematic iteration of a method that replicates the stored content of proxy assets into a more efficient data format.

79. The apparatus of any of claims 75 to 78 incorporating further means to progressively improve the interoperability of the digital library system by enabling systematic iterative replication of the stored content in alternative data formats.

80. The apparatus of claim 79 wherein the further means is a computer program configured to iteratively replicate the stored content of each of the proxy assets in a user-selected batch of proxy assets into an alternative data format selected by the user.

81. The apparatus of claim 79 wherein the further means is a computer program and user interface configured to support users in the systematic iteration of a method that replicates the stored content of proxy assets into an alternative data format.

82. The apparatus of any of claims 54 to 81 wherein the data processing means is configured to enable alteration of the metadata of the given proxy asset.

83. The apparatus of claim 82 wherein the metadata alteration means incorporates means to edit the metadata in a way that increases its quality.

84. The apparatus of claim 83 incorporating further means to progressively improve the effectiveness of the digital library's search means by enabling the making of systematic iterative improvements to the quality of the metadata contained within any of the proxy assets.

85. The apparatus of claim 84 wherein the further means is a computer program configured to iteratively improve the quality of the metadata of each of the proxy assets in a user-selected batch of proxy assets.

86. The apparatus of claim 84 wherein the further means is a computer program and user interface configured to support users in the systematic iteration of a method that improves the quality of the metadata of proxy assets.

87. The apparatus of any of claims 82 to 86 wherein the metadata alteration means incorporates means to increase the amount of metadata describing an asset.

88. The apparatus of claim 87 incorporating further means to progressively improve the versatility of the digital library's search means by enabling the making of systematic iterative additions to the metadata contained within any of the proxy assets.

89. The apparatus of claim 88 wherein the means is a computer program configured to iteratively add further metadata to each of the proxy assets in a user-selected batch of proxy assets.

90. The apparatus of claim 88 wherein the further means is a computer program and user interface configured to support users in the systematic iteration of a method that adds further metadata to proxy assets.

91. The apparatus of any of claims 82 to 90 wherein the metadata alteration means incorporates means to convert a themed proxy asset that satisfies appropriate criteria into a logical proxy asset.

92. The apparatus of any of claims 82 to 91 wherein the metadata alteration means incorporates means to add by reference one or more additional data portions to the given proxy asset.

93. The apparatus of any of claims 82 to 92 wherein the metadata alteration means incorporates means to modify the given proxy asset's references to data portions in order to de-reference one or more data portions from the given proxy asset.

94. The apparatus of any previous claim wherein the digital library is a component of or interacts with one or several other electronic information systems or subsystems such as document management systems, enterprise content management systems, multimedia information capture systems, other digital library systems, on-line bookstores, or mixtures of such further systems.

95. A method for using the digital library system of any of claims 1 to 94 to perform the functions and operations enabled by the subsystems, parts and features of the apparatus of claims any of 1 to 94.

96. A computer program encoding computer-executable means that, when executed by a computer system, causes the system to perform the electronic processes that enable the methods of a digital library, said program incorporating program code that enables execution of the method of claim 95.

97. A computer readable storage medium having stored thereon computer readable program code, which, when executed by a computer system, causes the computer system to perform the processes that enable the methods of a digital library system incorporating the method of claim 95.

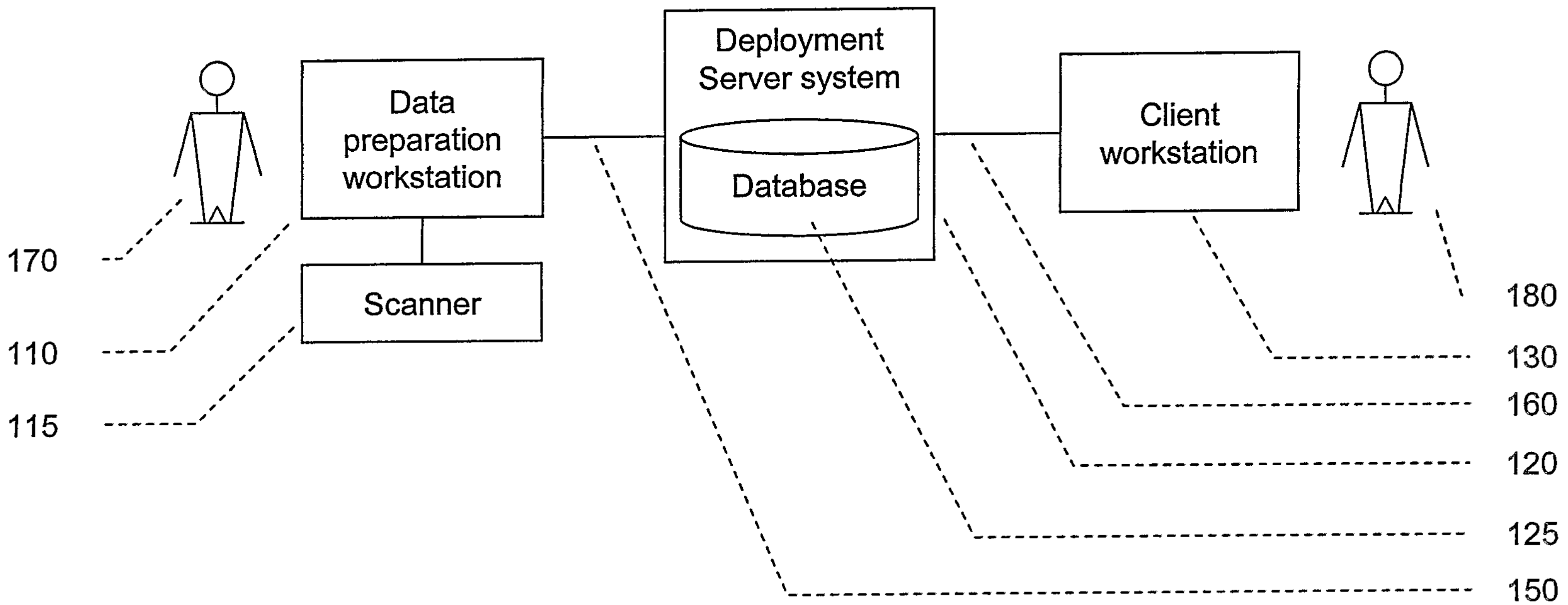

Data preparation workstation
Scanner
Deployment Server system
Database
Client workstation
170
110
115
180
130
160
120
125
150

Figure 1

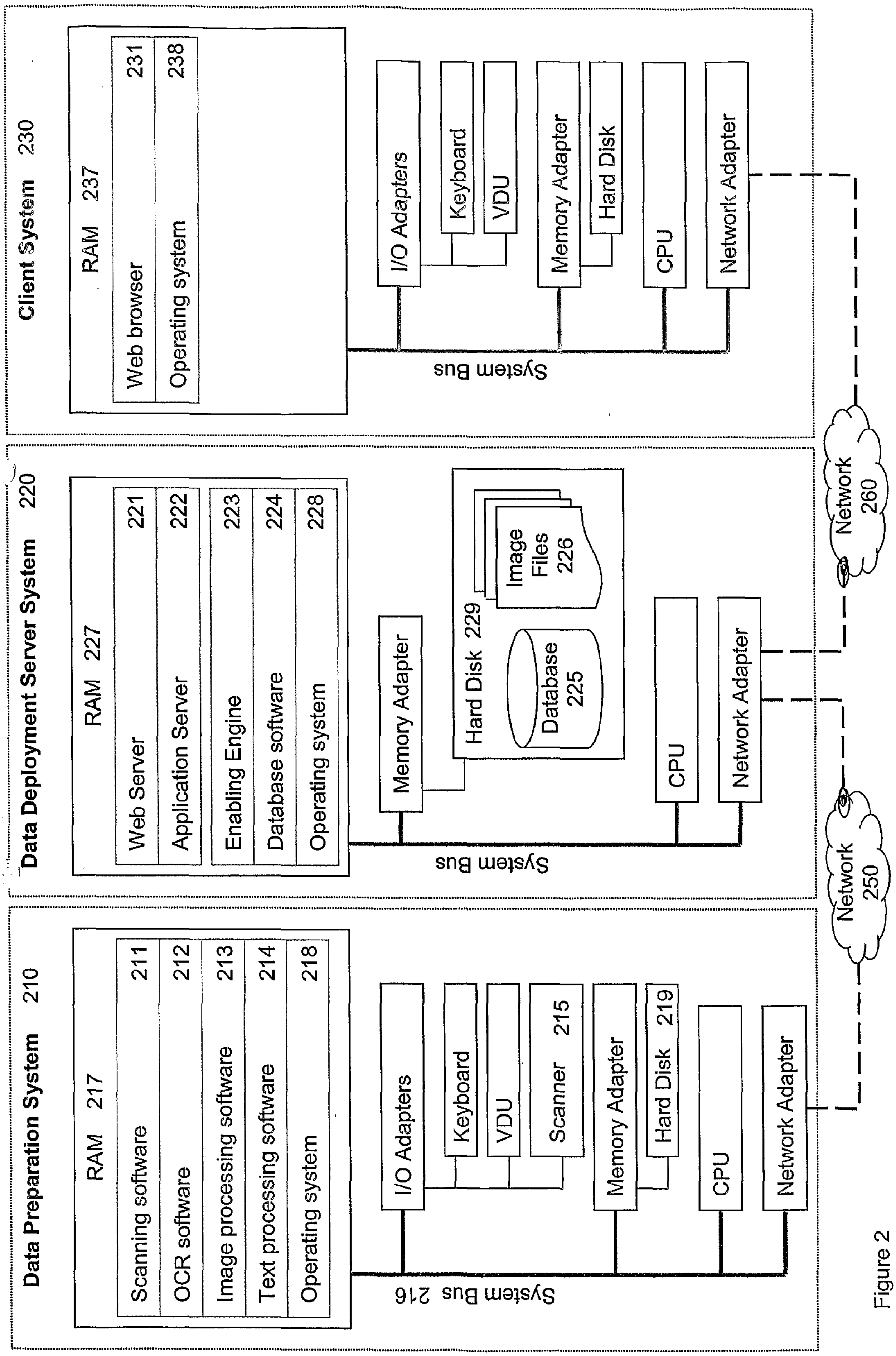

Data Preparation System 210
RAM 217
Scanning software 211
OCR software 212
Image processing software 213
Text processing software 214
Operating system 218
I/O Adapters
Keyboard
VDU
Scanner 215
Memory Adapter
Hard Disk 219
CPU
Network Adapter
System Bus 216
Data Deployment Server System 220
RAM 227
Web Server 221
Application Server 222
Enabling Engine 223
Database software 224
Operating system 228
Memory Adapter
Hard Disk 229
Database 225
Image Files 226
CPU
Network Adapter
System Bus
Client System 230
RAM 237
Web browser 231
Operating system 238
I/O Adapters
Keyboard
VDU
Memory Adapter
Hard Disk
CPU
Network Adapter
System Bus
Network 250
Network 260

Figure 2

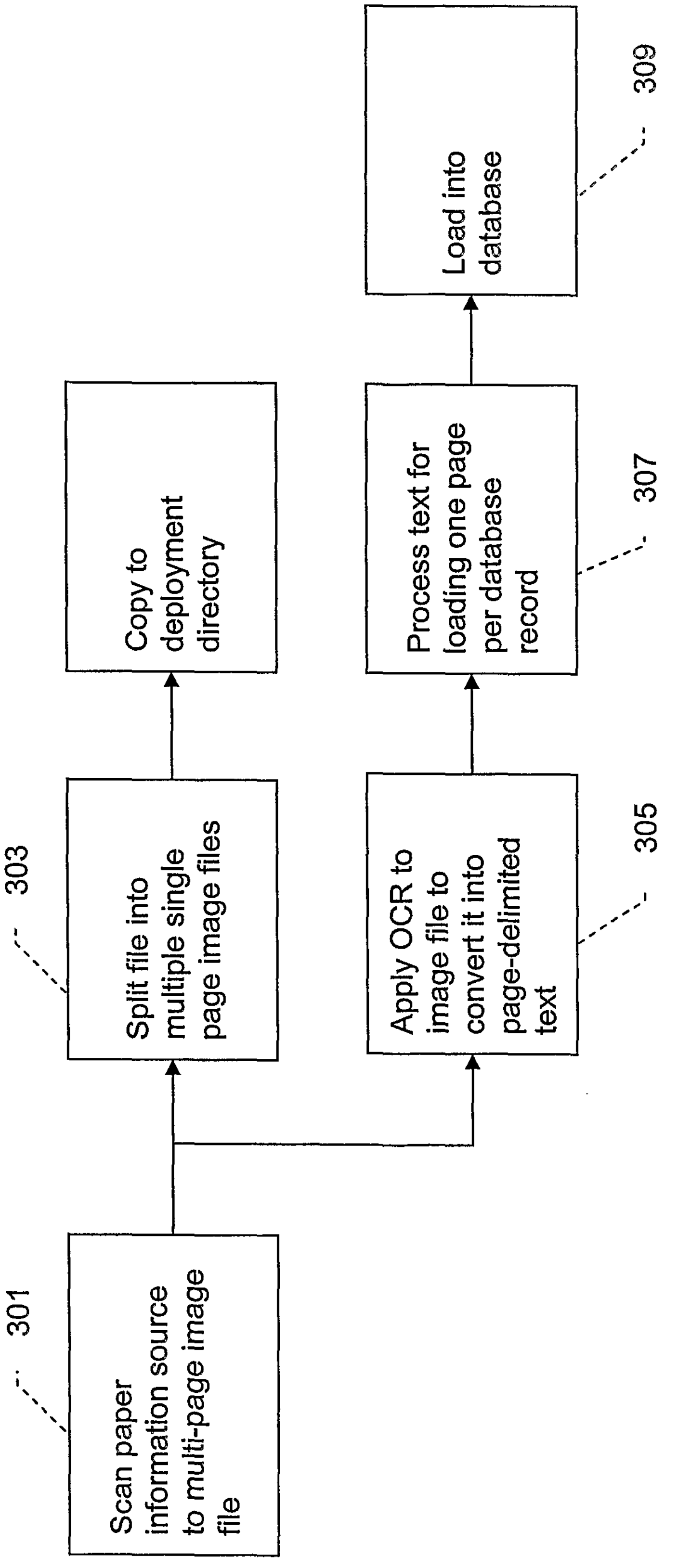

Scan paper information source to multi-page image file
301
Split file into multiple single page image files
303
Copy to deployment directory
Apply OCR to image file to convert it into page-delimited text
305
Process text for loading one page per database record
307
Load into database
309

Figure 3

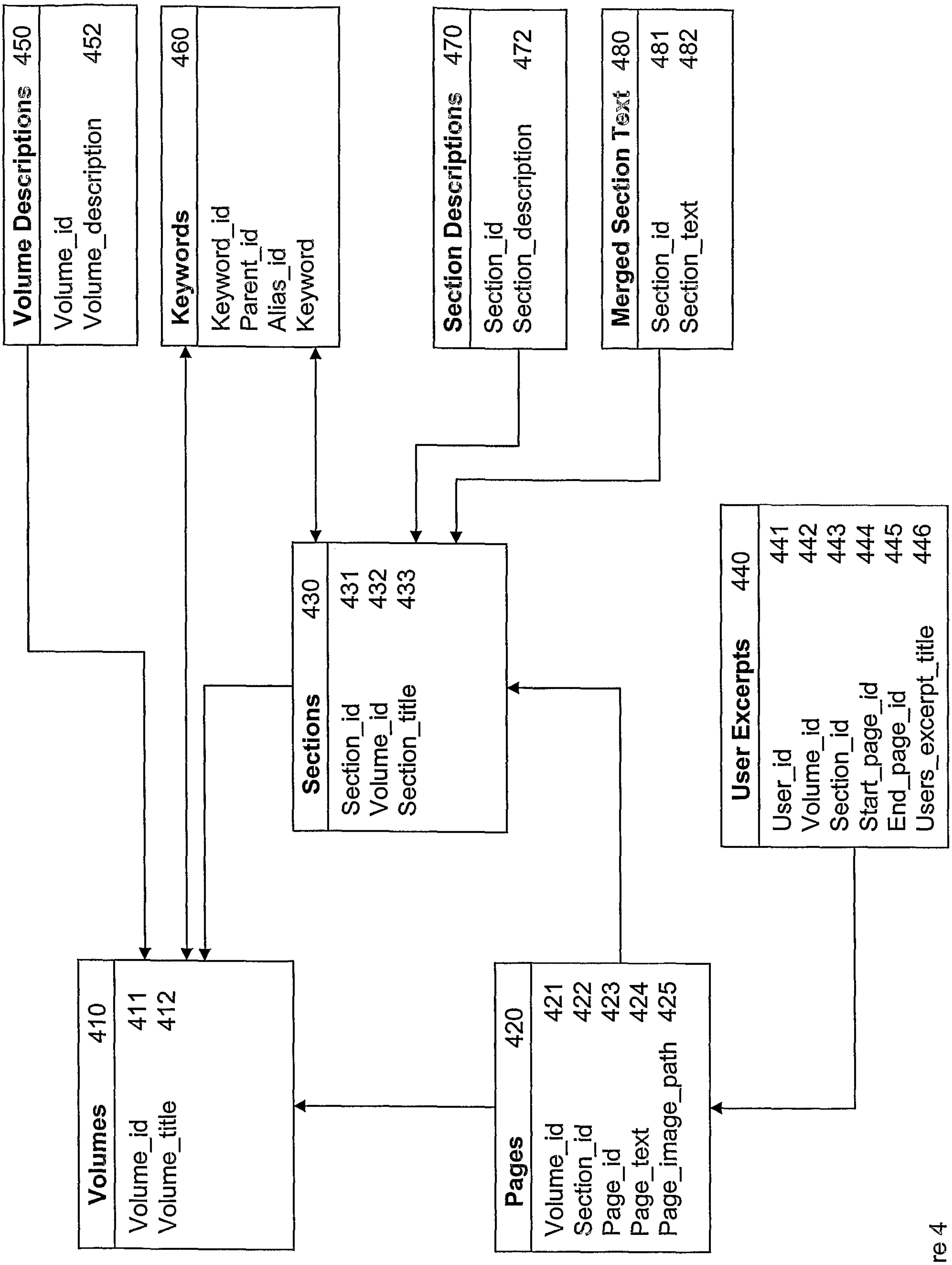

Volume Descriptions 450
Volume_id
Volume_description 452
Keywords 460
Keyword_id
Parent_id
Alias_id
Keyword
Section Descriptions 470
Section_id
Section_description 472
Merged Section Text 480
Section_id 481
Section_text 482
Sections 430
Section_id 431
Volume_id 432
Section_title 433
User Excerpts 440
User_id 441
Volume_id 442
Section_id 443
Start_page_id 444
End_page_id 445
Users_excerpt_title 446
Volumes 410
Volume_id 411
Volume_title 412
Pages 420
Volume_id 421
Section_id 422
Page_id 423
Page_text 424
Page_image_path 425

Figure 4

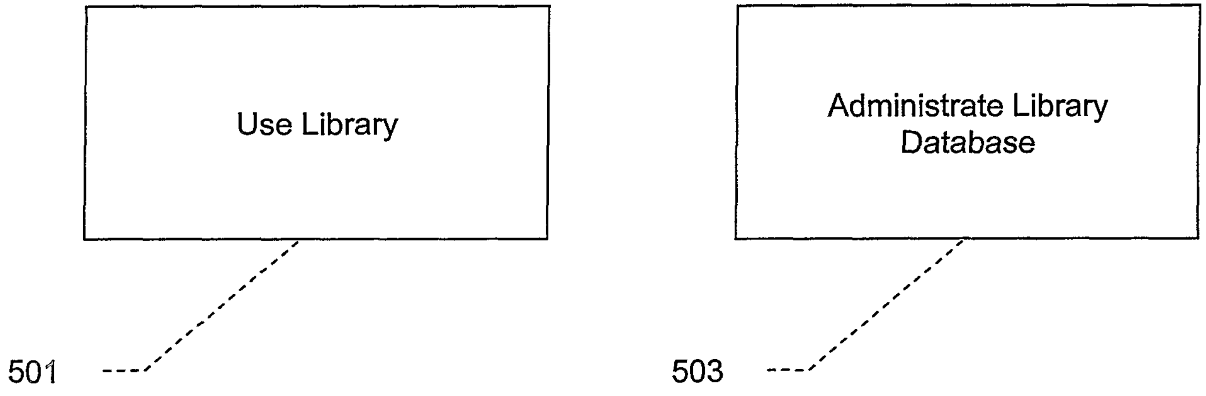


Figure 5

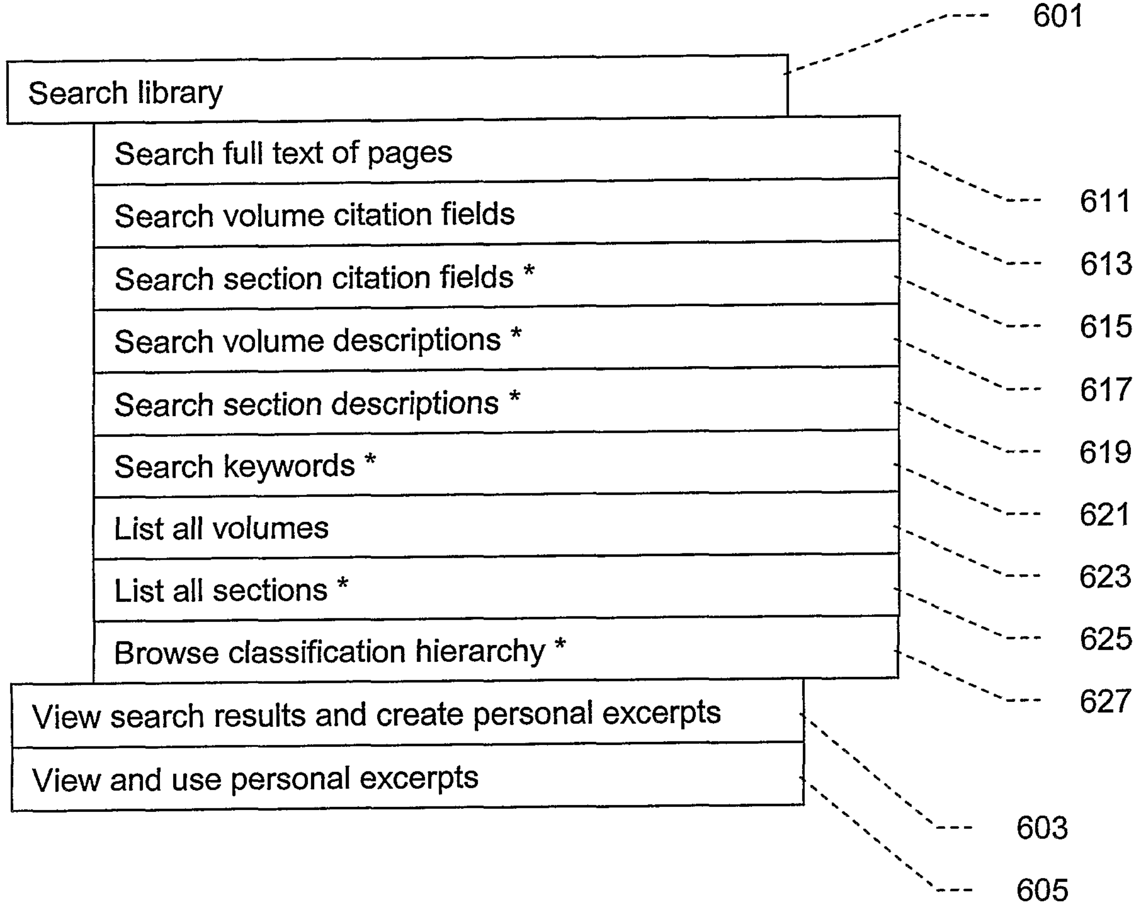

Search library
601
Search full text of pages
611
Search volume citation fields
613
Search section citation fields *
615
Search volume descriptions *
617
Search section descriptions *
619
Search keywords *
621
List all volumes
623
List all sections *
625
Browse classification hierarchy *
627
View search results and create personal excerpts
603
View and use personal excerpts
605

Figure 6

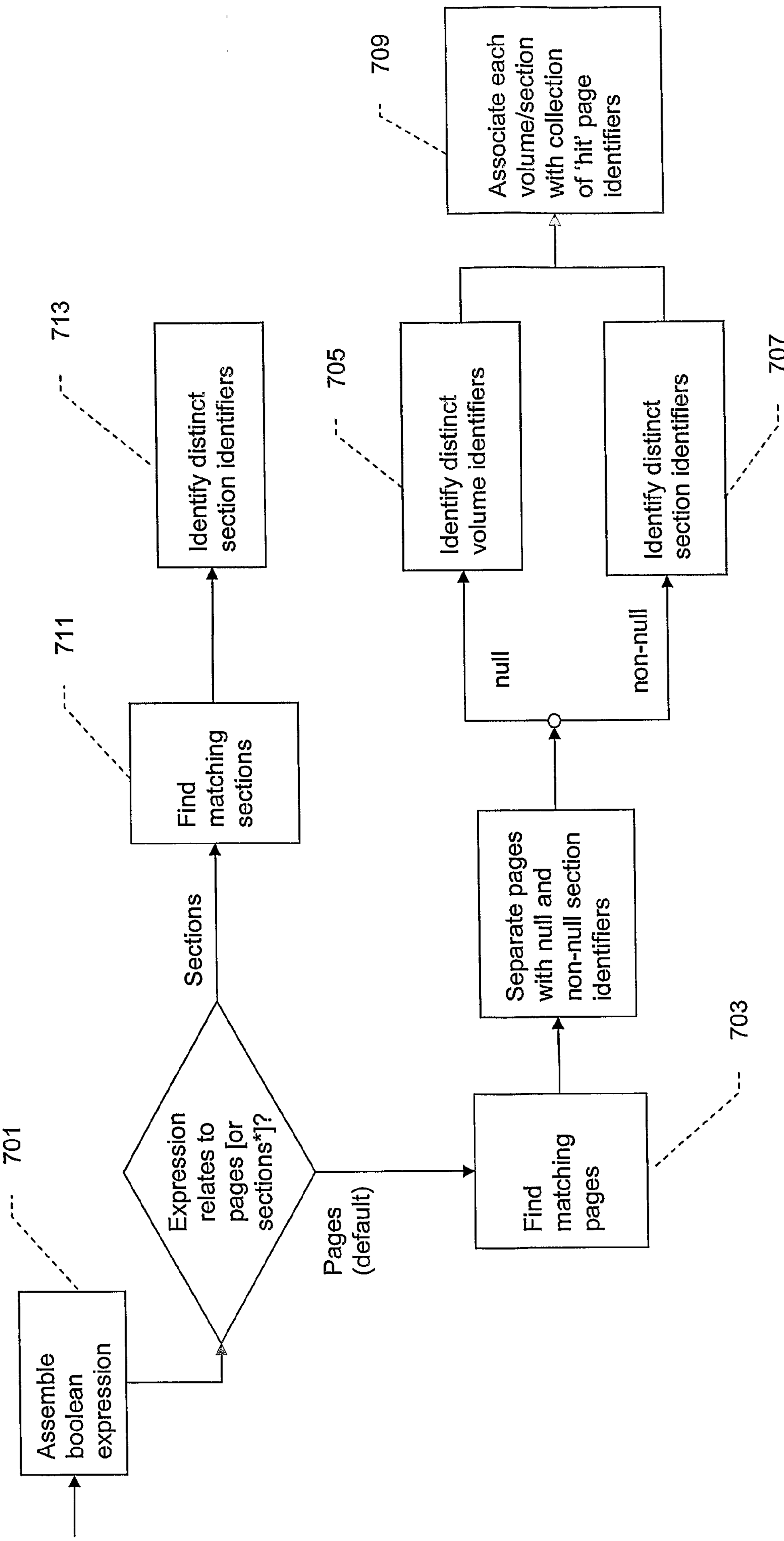

701
Assemble boolean expression
Expression relates to pages [or sections*]?
Sections
Pages (default)
711
Find matching sections
713
Identify distinct section identifiers
703
Find matching pages
Separate pages with null and non-null section identifiers
null
non-null
705
Identify distinct volume identifiers
707
Identify distinct section identifiers
709
Associate each volume/section with collection of 'hit' page identifiers

Figure 7

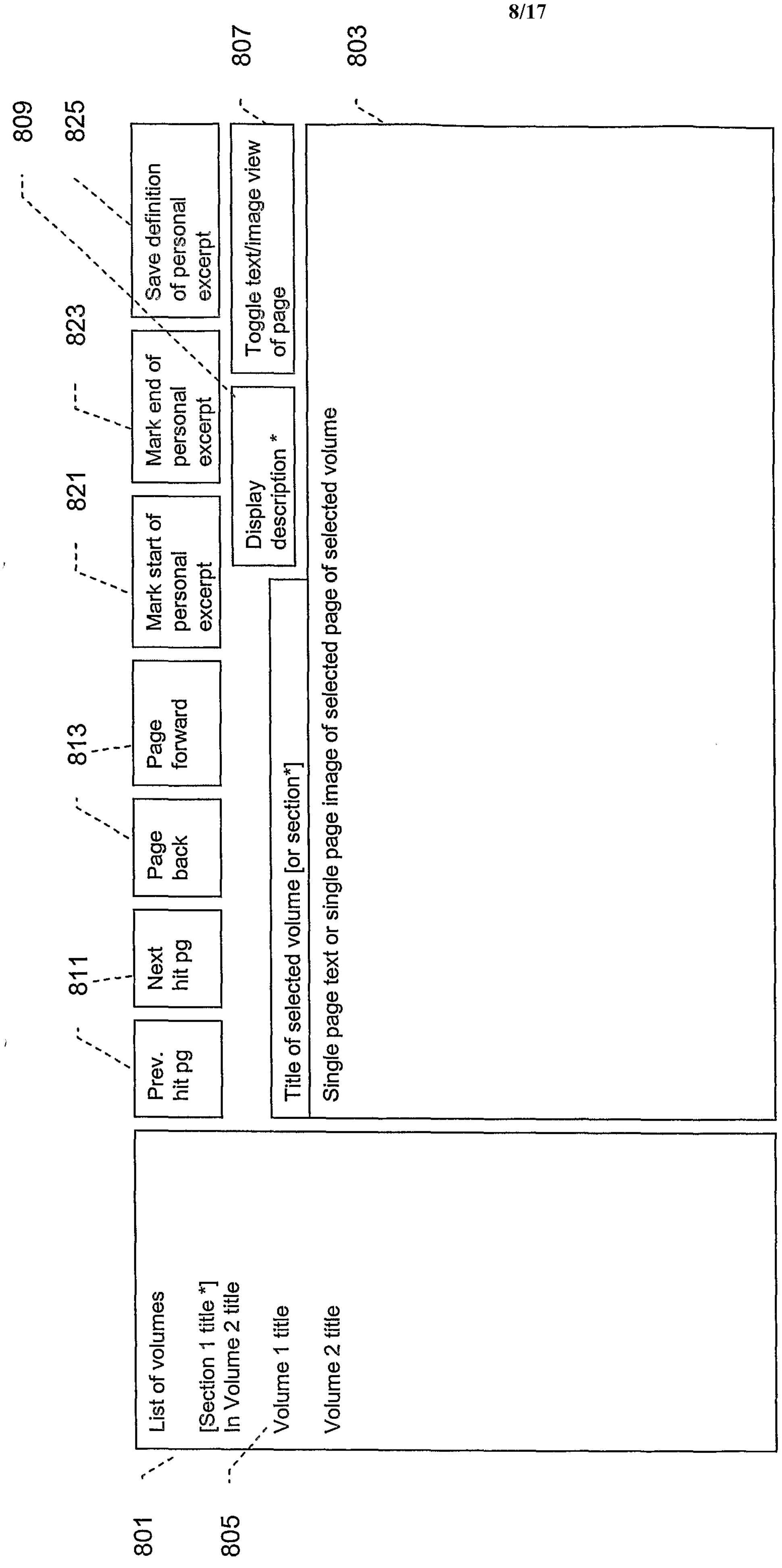

801
805
List of volumes
[Section 1 title *]
In Volume 2 title
Volume 1 title
Volume 2 title
Prev. hit pg
Next hit pg
811
Page back
Page forward
813
Mark start of personal excerpt
821
Mark end of personal excerpt
823
Save definition of personal excerpt
825
809
Title of selected volume [or section*]
Display description *
Toggle text/image view of page
807
Single page text or single page image of selected page of selected volume
803

Figure 8

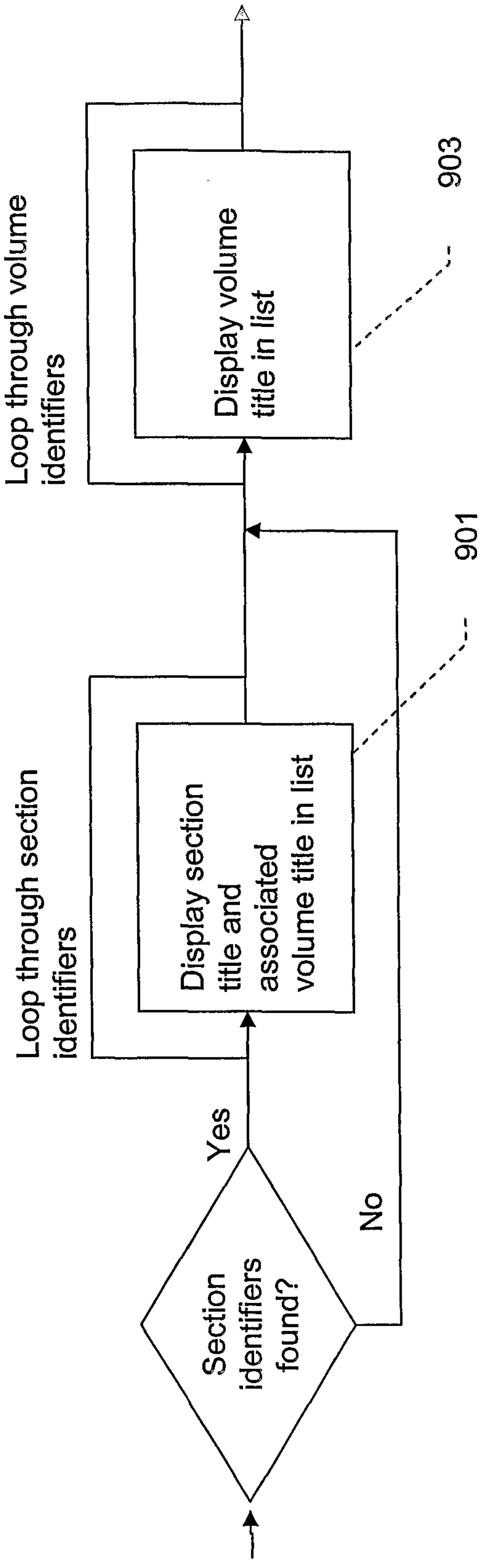

Section identifiers found?
Yes
No
Loop through section identifiers
Display section title and associated volume title in list
901
Loop through volume identifiers
Display volume title in list
903

Figure 9

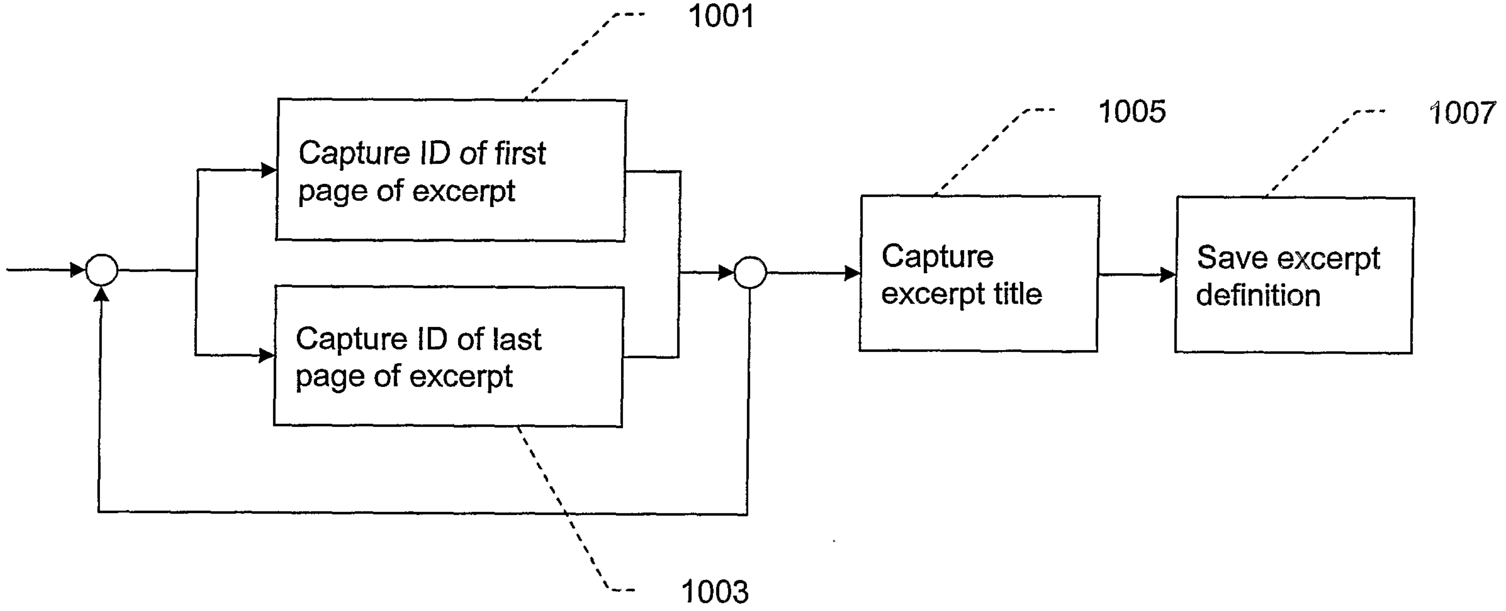

1001
Capture ID of first page of excerpt
Capture ID of last page of excerpt
1003
1005
Capture excerpt title
1007
Save excerpt definition

Figure 10

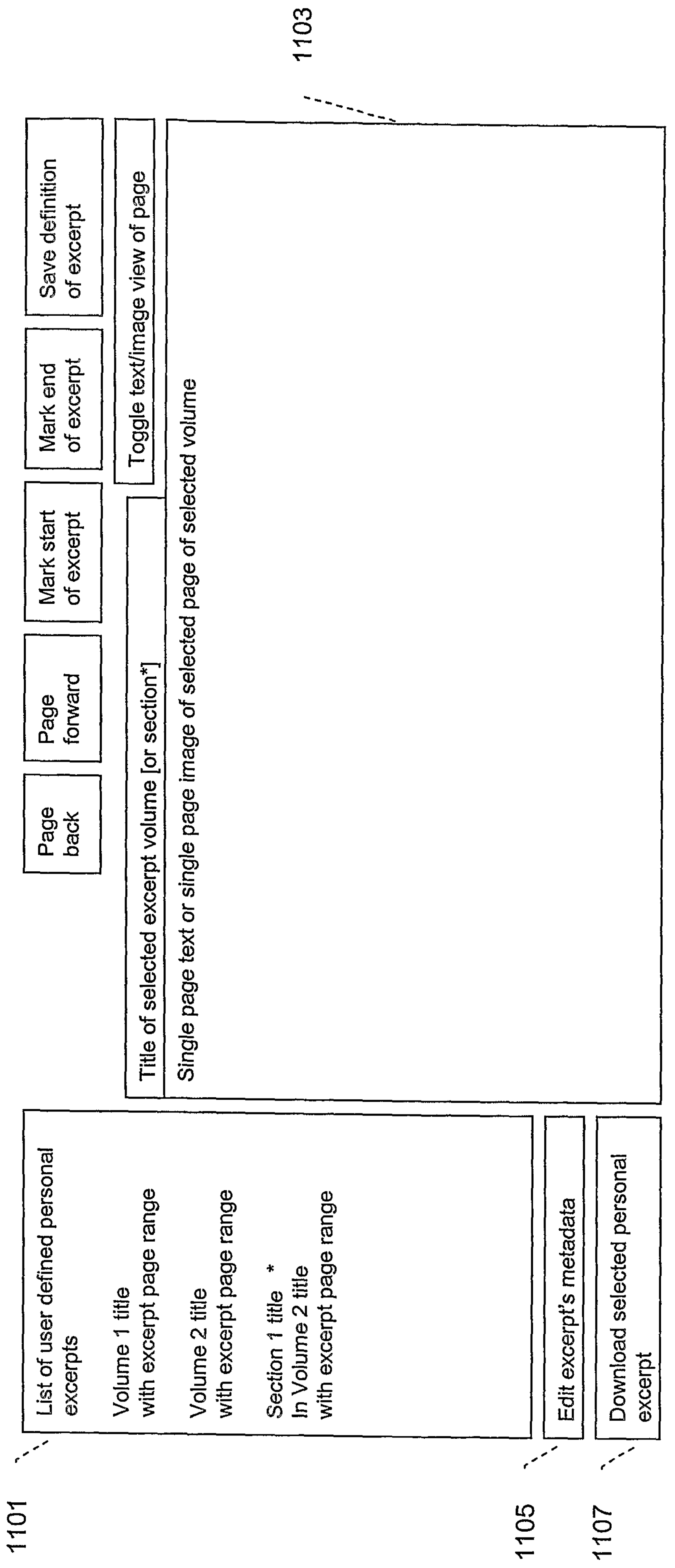

1103
Page back
Page forward
Mark start of excerpt
Mark end of excerpt
Save definition of excerpt
Toggle text/image view of page
Title of selected excerpt volume [or section*]
Single page text or single page image of selected page of selected volume
1101
List of user defined personal excerpts
Volume 1 title with excerpt page range
Volume 2 title with excerpt page range
Section 1 title * In Volume 2 title with excerpt page range
1105
Edit excerpt's metadata
1107
Download selected personal excerpt

Figure 11

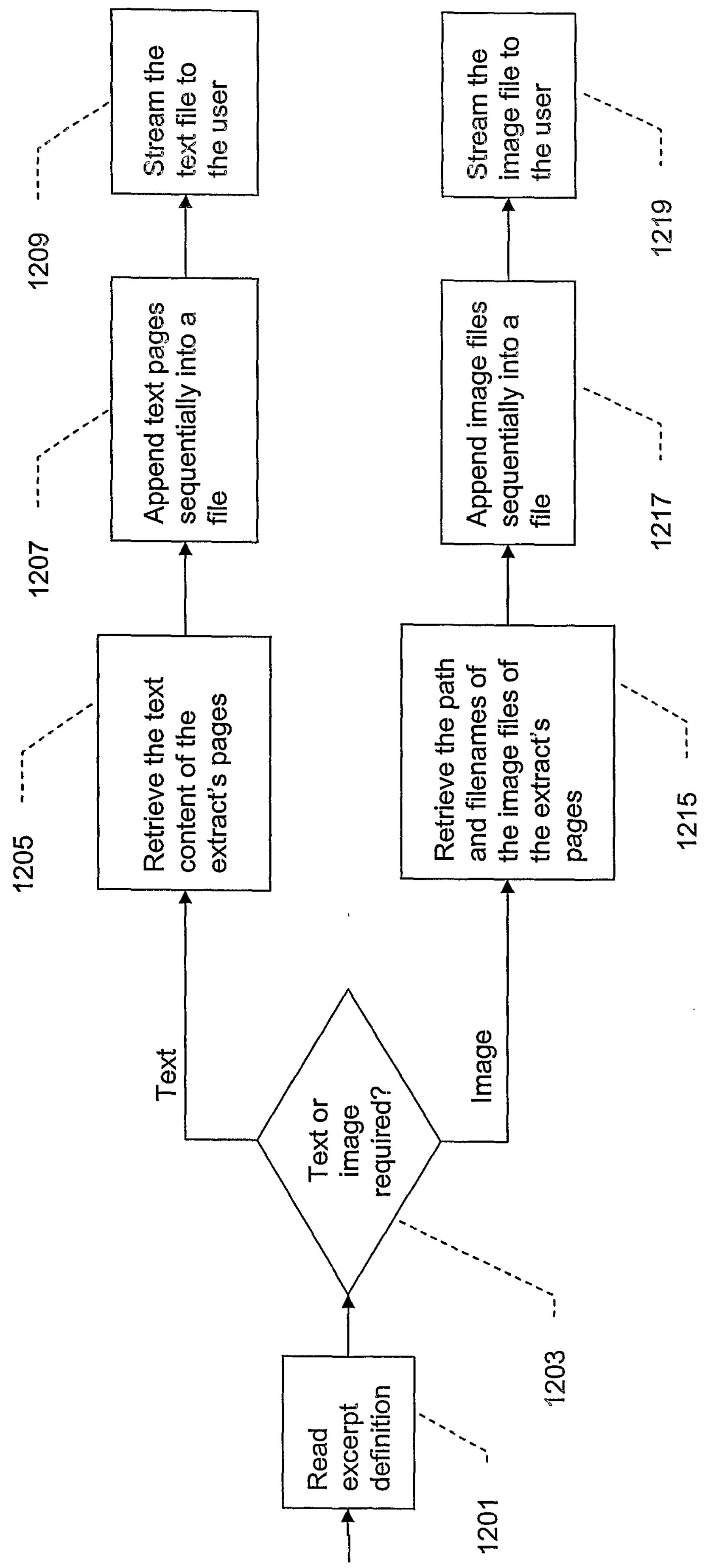

Read excerpt definition
1201
Text or image required?
1203
Text
Image
Retrieve the text content of the extract's pages
1205
Append text pages sequentially into a file
1207
Stream the text file to the user
1209
Retrieve the path and filenames of the image files of the extract's pages
1215
Append image files sequentially into a file
1217
Stream the image file to the user
1219

Figure 12

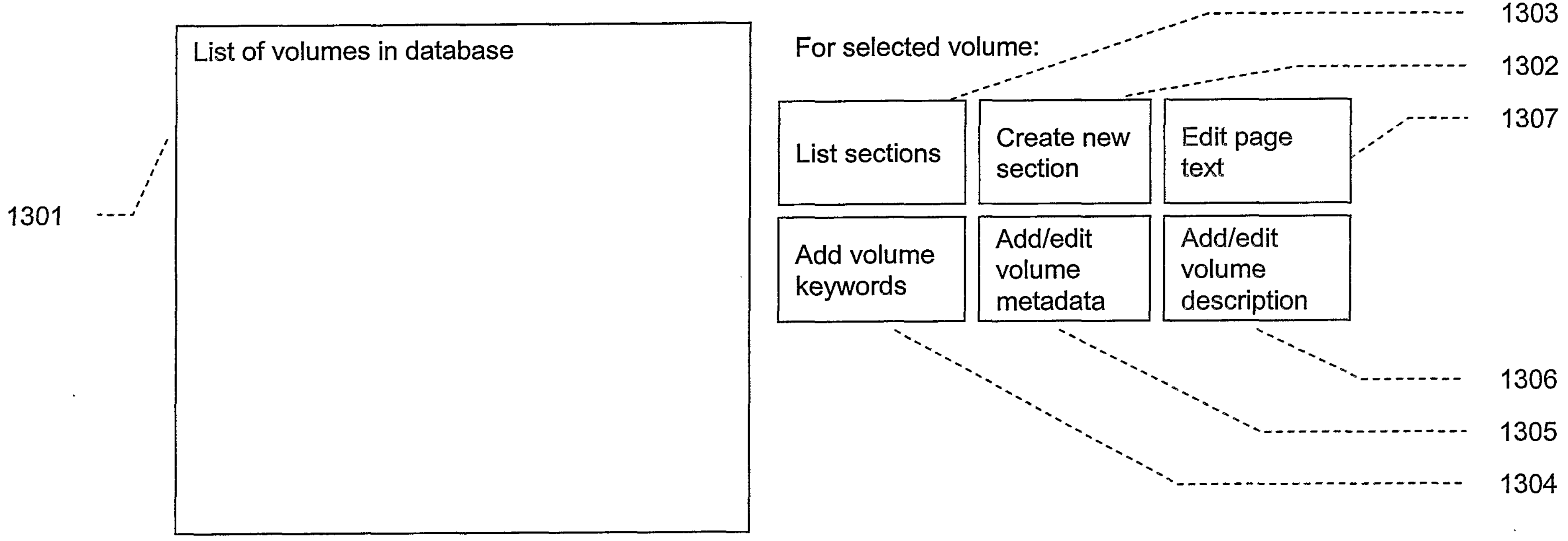

List of volumes in database
For selected volume:
List sections
Create new section
Edit page text
Add volume keywords
Add/edit volume metadata
Add/edit volume description
1301
1303
1302
1307
1306
1305
1304

Figure 13

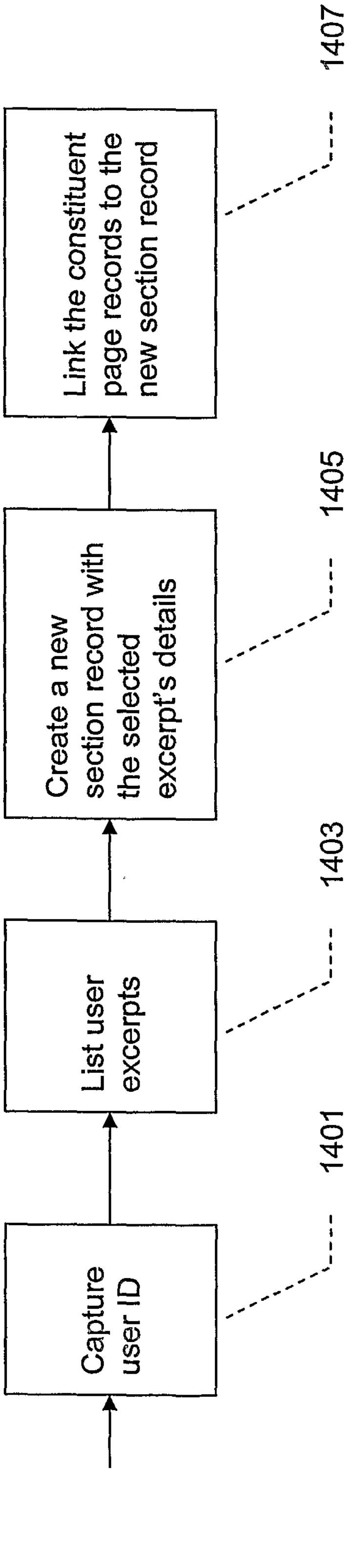

Capture user ID
1401
List user excerpts
1403
Create a new section record with the selected excerpt's details
1405
Link the constituent page records to the new section record
1407

Figure 14

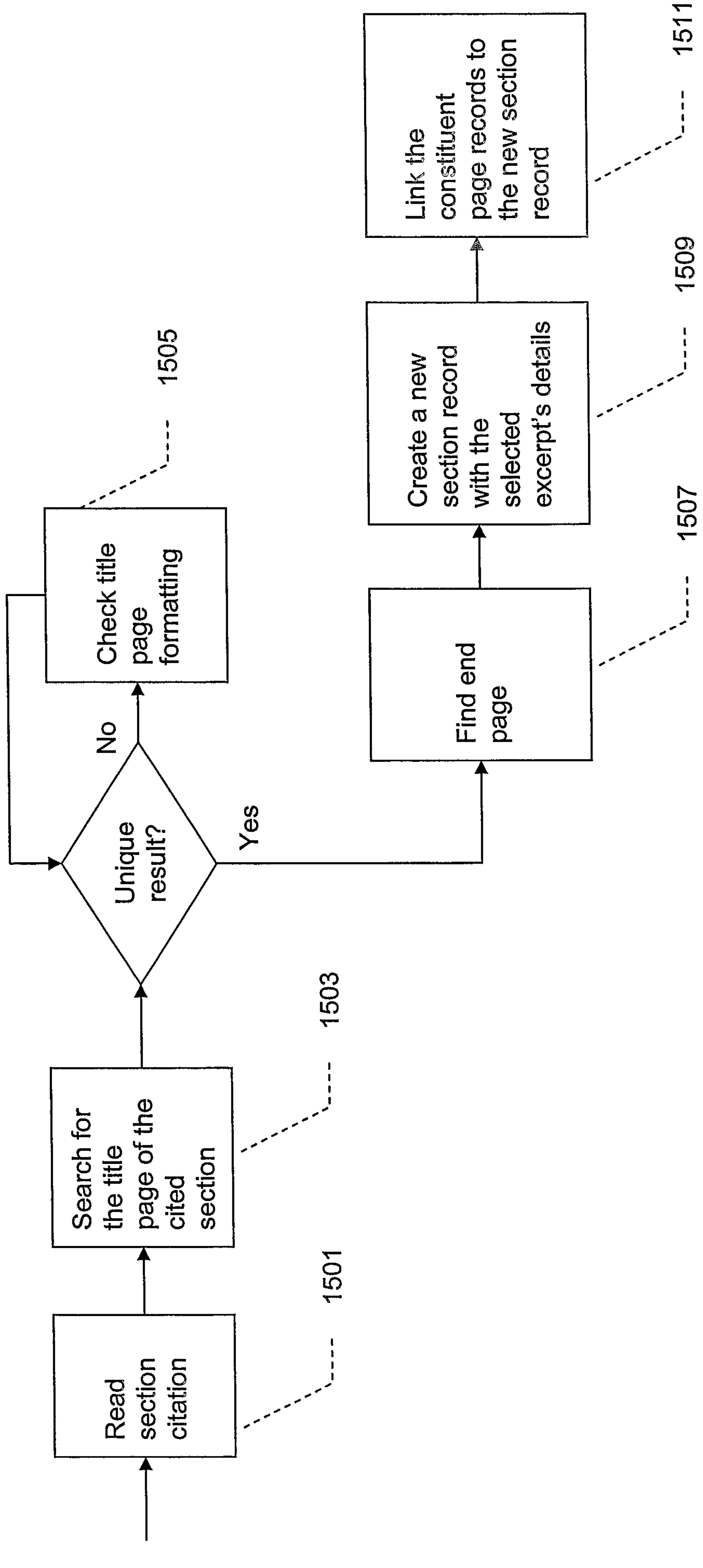

Read section citation
1501
Search for the title page of the cited section
1503
Unique result?
No
Check title page formatting
1505
Yes
Find end page
1507
Create a new section record with the selected excerpt's details
1509
Link the constituent page records to the new section record
1511

Figure 15

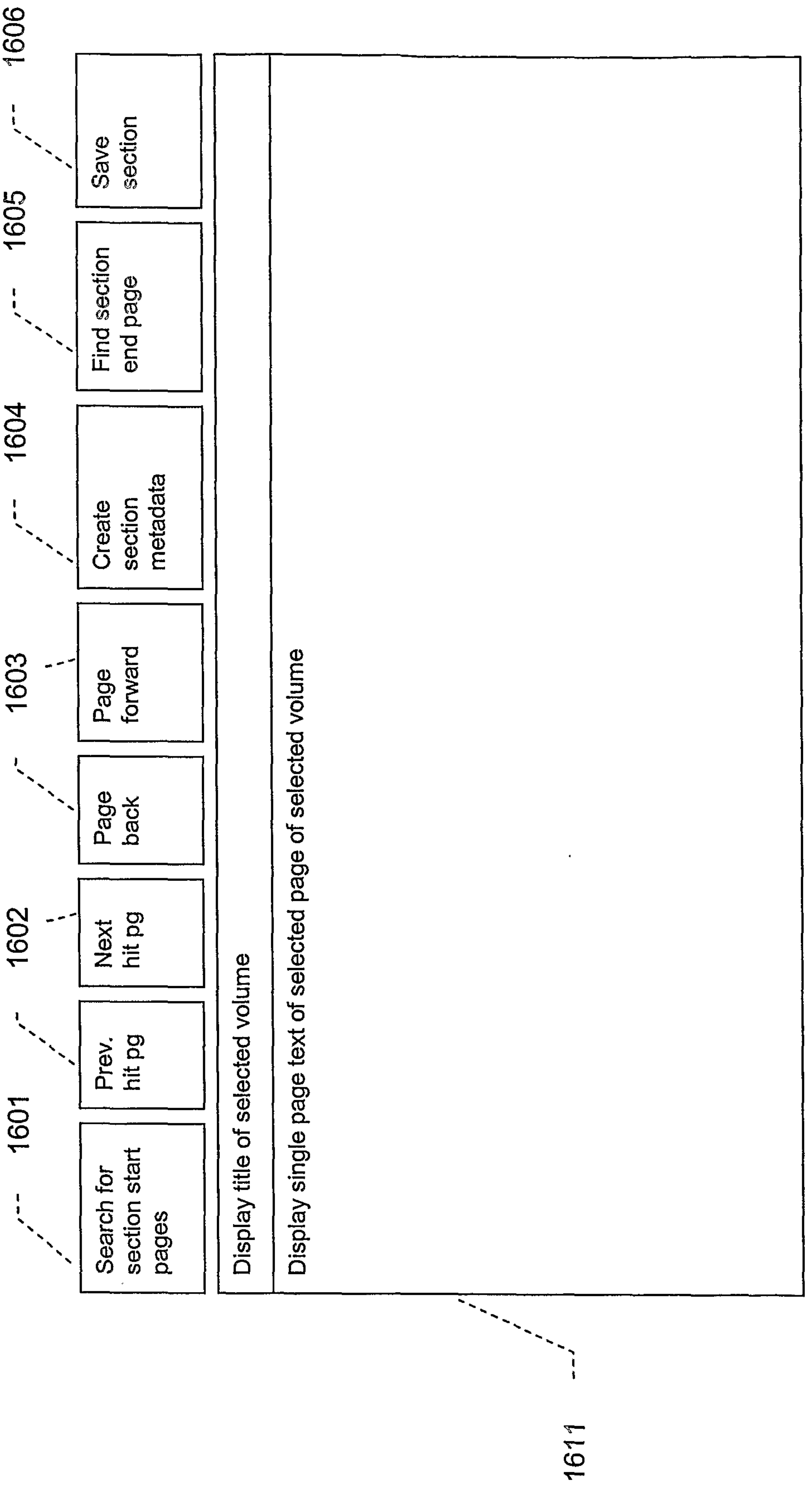

1601
1602
1603
1604
1605
1606
Search for section start pages
Prev. hit pg
Next hit pg
Page back
Page forward
Create section metadata
Find section end page
Save section
Display title of selected volume
Display single page text of selected page of selected volume
1611

Figure 16

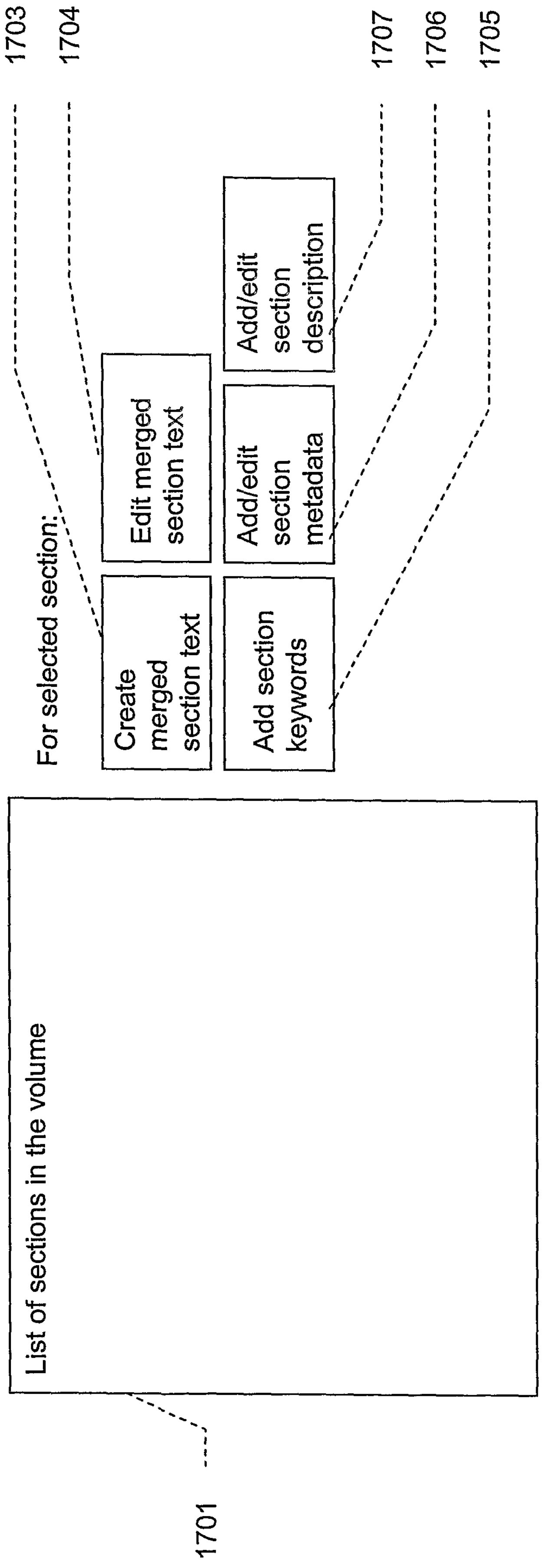

1703
1704
1707
1706
1705
List of sections in the volume
For selected section:
Create merged section text
Edit merged section text
Add section keywords
Add/edit section metadata
Add/edit section description
1701

Figure 17

INTERNATIONAL SEARCH REPORT

ional Application No
GB2004/001912

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2002/052861 A1 (GUSTMAN SAMUEL) 2 May 2002 (2002-05-02)	1-31, 36-42, 44-74, 82-97
Y	the whole document	32-35, 75-81
X	US 6 076 091 A (GREEF ARTHUR REGINALD ET AL) 13 June 2000 (2000-06-13) column 3, line 32 - column 12, line 14; claims	1,10, 19-24, 29,37-44
	-/--	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search 7 September 2004	Date of mailing of the international search report 16/09/2004
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Authorized officer Herry, T

INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB2004/001912

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 00/02143 A (FOX HARRY ; BECKER SHLOMIT (IL); BENJAMIN JACOB (IL); BRODY DANIEL (IL) 13 January 2000 (2000-01-13) page 8 - page 25 page 185 page 211; claims -----	1-21, 32-35,43
Y	US 2002/152267 A1 (LENNON ALISON J) 17 October 2002 (2002-10-17) paragraph ‘0210! - paragraph ‘0226! -----	32-35, 75-81
X	US 5 832 499 A (GUSTMAN SAMUEL) 3 November 1998 (1998-11-03)	1
A	column 12 - column 14 -----	2-97

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
GB2004/001912

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2002052861 A1	02-05-2002	US 6212527 B1	03-04-2001
US 6076091 A	13-06-2000	NONE	
WO 0002143 A	13-01-2000	AU 4644899 A	24-01-2000
		CA 2336715 A1	13-01-2000
		EP 1097422 A1	09-05-2001
		WO 0002143 A1	13-01-2000
US 2002152267 A1	17-10-2002	AU 770181 B2	12-02-2004
		AU 9737701 A	27-06-2002
US 5832499 A	03-11-1998	AU 3880497 A	02-02-1998
		EP 0912951 A2	06-05-1999
		WO 9801849 A2	15-01-1998
		US 6092080 A	18-07-2000