

(21) Application No: 1901830.8

(22) Date of Filing: 08.02.2019

(71) Applicant(s):  
All Street Research Limited  
1 Phipp Street, Shoreditch, London, EC2A4PS,  
United Kingdom

(72) Inventor(s):  
Matthew Karas

(74) Agent and/or Address for Service:  
Reddie & Grose LLP  
The White Chapel Building,  
10 Whitechapel High Street, London, E1 8QS,  
United Kingdom

(51) INT CL:  
G06F 16/16 (2019.01) G06F 16/93 (2019.01)

(56) Documents Cited:  
WO 2010/014403 A1 US 20100030763 A1  
US 20080098317 A1 US 20060218034 A1

(58) Field of Search:  
INT CL G06F  
Other: WPI, EPODOC, Patent Fulltext

(54) Title of the Invention: **Method and system for capturing metadata in a document object or file format**  
Abstract Title: **Method and system for capturing metadata in a document object or file format**

(57) Capturing and preserving metadata in a document object or file format by presenting via a first user interface controls for a user to interact with a document database. Presenting, via a second user interface, controls for a user to enter and record text-based input into a document object or file format. Capturing as metadata the user's interactions with the document database and adding it to the document object or file format. Capturing input text entered by the user and adding it to the document object or file format and determining an association between the text entered by the user and the captured metadata then updating the metadata to reflect the association. Capturing an editing action from the user modifying text already entered by the user, adding metadata to the document object or file format to identify the modifications made to the text by the editing action and saving the document object or file format, including the text entered by the user, the modifications made to the text by the editing action, and the metadata capturing the user interactions and the determined association.

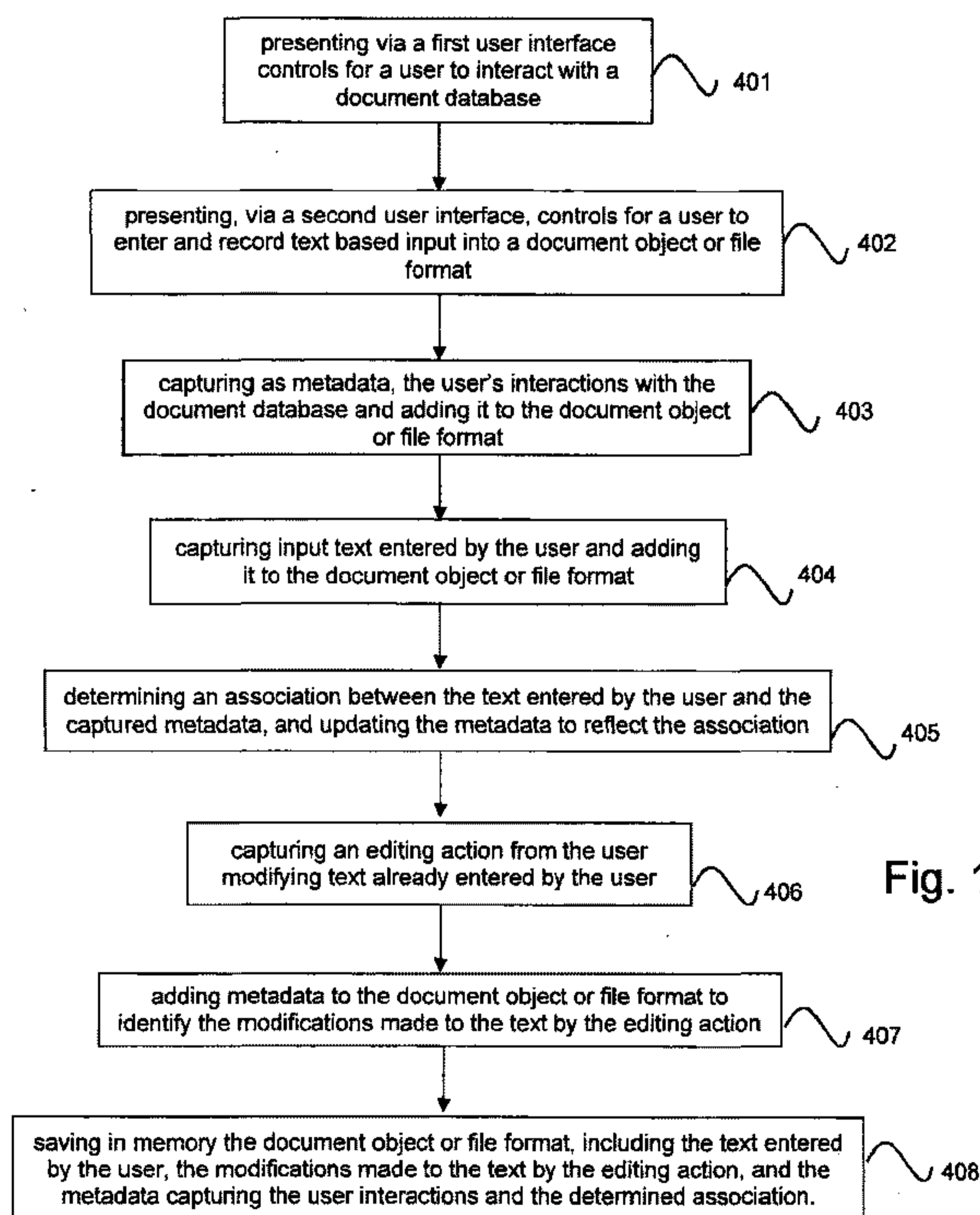


Fig. 10

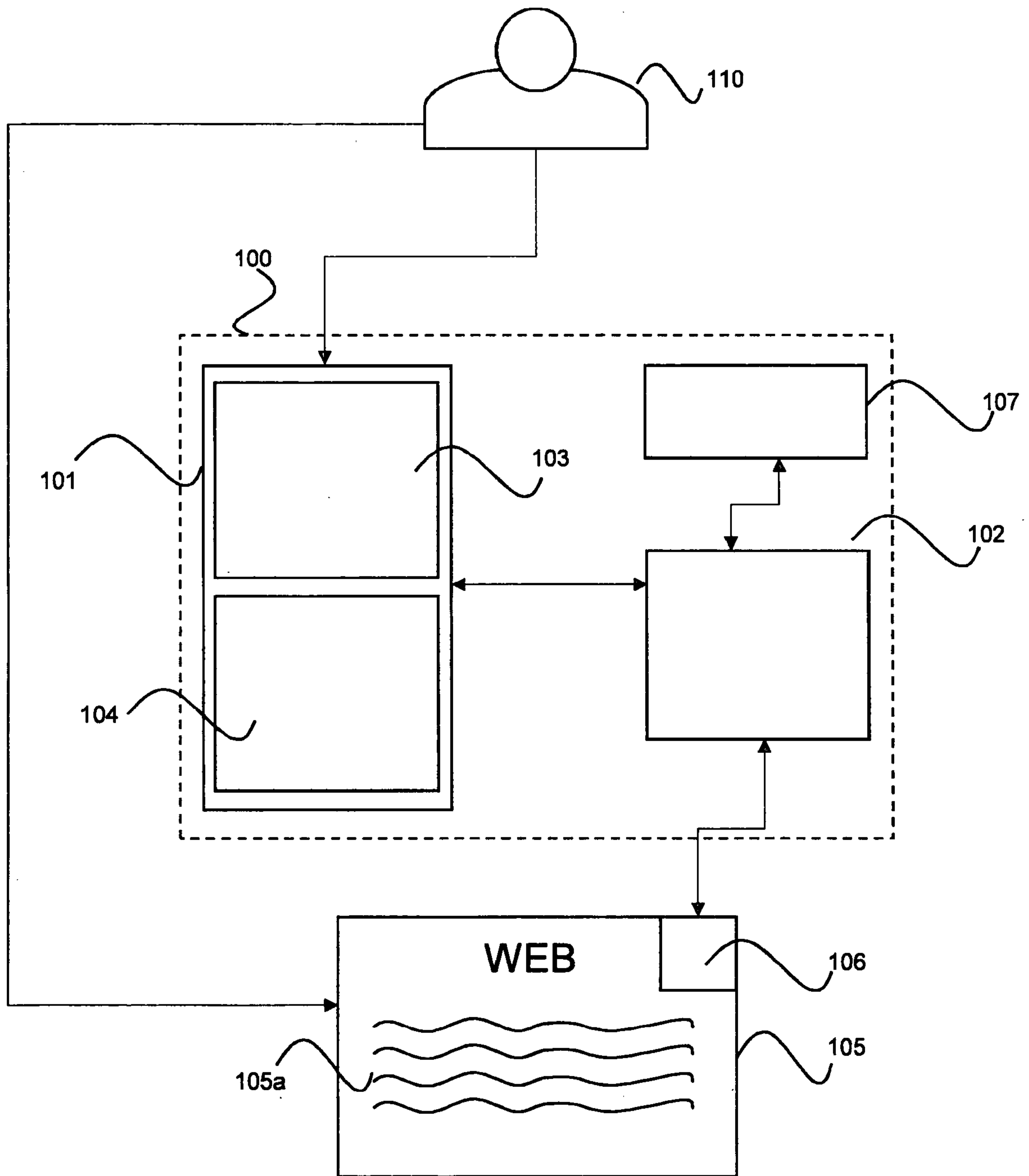


Fig. 1

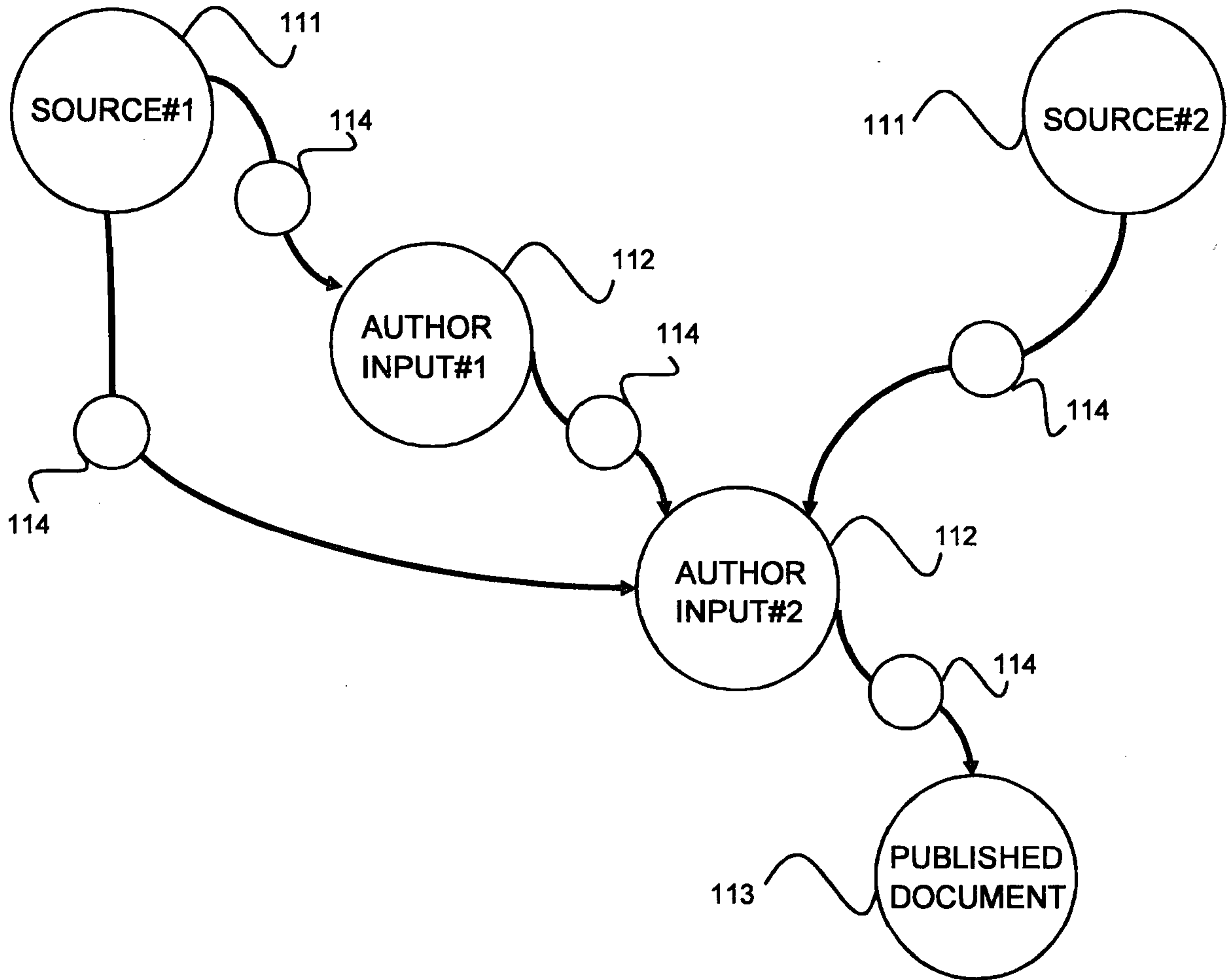


Fig. 2




301 	302 	303 
#1	The opportunities for Artificial Intelligence in health care /r	ID#27 / 12.56pm Manual Input

Fig. 3

#1	The opportunities for Artificial Intelligence in health care /r /r	ID#27 / 12.56pm Manual Input
#2	Artificial Intelligence provides many opportunities for improving healthcare, particularly with application to medical diagnostics. Medical staff typically diagnose a patient's underlying health issues, based on a visual inspection of a patient medical data, such as MRI or EKG data. Diagnosis is based on the ability of the medical professional, their experience, and sometimes luck in spotting the tell-tale signs of disease. Some conditions may be more difficult to detect than others.	ID#27 / 1.02pm Copy from: <link #1>
#3	Machine Learning algorithms, such as those provided in the field of Artificial Intelligence, can also be trained using real medical data, benefitting from the collective experience of the medical professionals involved in the training. Such algorithms can typically extrapolate that training better than humans, reducing the element of luck in the diagnosis. Machine learning algorithms provided by Medical AI Ltd have shown to be 20% more reliable than doctors in actual field trials.	ID#27 / 1.05pm Copy from: <link #2>
#4	-----	<link #3> 1.06pm browsed

Fig. 4

#1	The opportunities for Artificial Intelligence in health care /r /r	ID#27 / 12.56pm Manual Input
#2	Artificial Intelligence provides many opportunities for improving	ID#27 / 1.02pm Copy from: <link #1> [unedited]
#3	<del>healthcare, particularly with application to medical diagnostics-</del> [in which] M[m]edical staff typically diagnose a patient's underlying health issues, based on a visual inspection of a patient medical data, <del>such as MRI or EKG data.</del>	ID#27 / 1.07pm Editing of #2 <link #1> edited
#4	Diagnosis is based on the ability of the medical professional, their experience, and sometimes luck in spotting the tell-tale signs of disease. Some conditions may be more difficult to detect than others.	ID#27 / 1.07pm Copy from: <link #1> [unedited]
#5	Machine Learning algorithms, such as those provided in the field of Artificial Intelligence, can also be trained using real medical data, benefitting from the collective experience of the medical professionals involved in the training. Such algorithms can typically extrapolate that training better than humans, reducing the element of luck in the diagnosis. Machine learning algorithms provided by Medical AI Ltd have shown to be 20% more reliable than doctors in actual field trials.	ID#27 / 1.05pm Copy from: <link #1>
#6	-----	<link #3> 1.06pm browsed

Fig. 5

#1	The opportunities for Artificial Intelligence in health care /r /r	ID#27 / 12.56pm Manual Input
#2	Artificial Intelligence provides many opportunities for improving	ID#27 / 1.02pm Copy from: <link #1> [unedited]
#3	<del>healthcare, particularly with application to medical diagnostics.</del> <u>[in which]</u> <del>M[m]</del> medical staff typically diagnose a patient's underlying health issues, based on a visual inspection of a patient medical data, such as MRI or EKG data.	ID#27 / 1.07pm Editing of #2 <link #1> edited
#4	Diagnosis is based on the ability of the medical professional, their experience, and sometimes luck in spotting the tell-tale signs of disease.	ID#27 / 1.07pm Copy from: <link #1> [unedited]
#5	<del>Some conditions may be more difficult to detect than others.</del>	ID#27 / 1.08pm Copy from: <link #1> [edited]
#6	Machine Learning algorithms, such as those provided in the field of Artificial Intelligence, can also be trained using real medical data, benefitting from the collective experience of the medical professionals involved in the training. Such algorithms can typically extrapolate that training better than humans, reducing the element of luck in the diagnosis. Machine learning algorithms provided by Medical AI Ltd have shown to be 20% more reliable than doctors in actual field trials.	ID#27 / 1.05pm Copy from: <link #2>
#7	-----	<link #3> 1.06pm browsed

Fig. 6

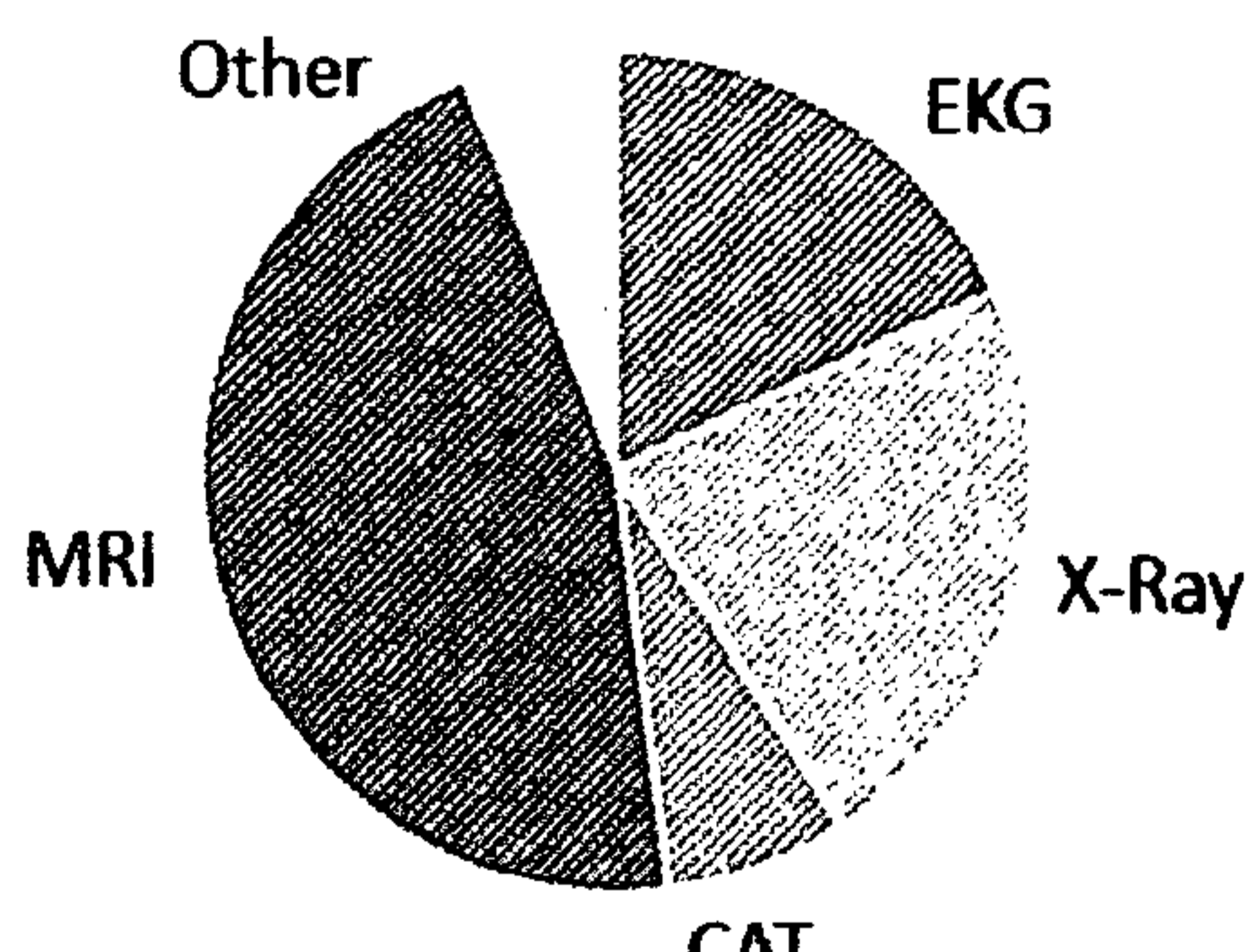
#1	The opportunities for Artificial Intelligence in health care /r /r	ID#27 / 12.56pm Manual Input												
#2	Artificial Intelligence provides many opportunities for improving	ID#27 / 1.02pm Copy from: <link> [unedited]												
#3	<del>healthcare, particularly with application to medical diagnostics.</del> [in which] M[m]edical staff typically diagnose a patient's underlying health issues, based on a visual inspection of a patient medical data, such as MRI or EKG data.	ID#27 / 1.07pm Editing of #2 <link #1> edited												
#4	Diagnosis is based on the ability of the medical professional, their experience, and sometimes luck in spotting the tell-tale signs of disease.	ID#27 / 1.07pm Copy from: <link #1> [unedited]												
#6	<del>Some conditions may be more difficult to detect than others.</del>	ID#27 / 1.08pm Copy from: <link #1> [edited]												
#7	Machine Learning algorithms, such as those provided in the field of Artificial Intelligence, can also be trained using real medical data, benefitting from the collective experience of the medical professionals involved in the training. Such algorithms can typically extrapolate that training better than humans, reducing the element of luck in the diagnosis. Machine learning algorithms provided by Medical AI Ltd have shown to be 20% more reliable than doctors in actual field trials.	ID#27 / 1.05pm Copy from: <link #2>												
#8	-----	<link #3> 1.06pm browsed												
#9	<p style="text-align: center;"><b>AI APPLICATIONS</b></p>  <p>A pie chart titled "AI APPLICATIONS" showing the distribution of AI applications across five categories: MRI, CAT, X-Ray, EKG, and Other. MRI is the largest slice, followed by CAT, X-Ray, EKG, and Other.</p> <table border="1"> <caption>AI Applications Data</caption> <thead> <tr> <th>Application</th> <th>Relative Size (Estimated)</th> </tr> </thead> <tbody> <tr> <td>MRI</td> <td>35%</td> </tr> <tr> <td>CAT</td> <td>25%</td> </tr> <tr> <td>X-Ray</td> <td>15%</td> </tr> <tr> <td>EKG</td> <td>10%</td> </tr> <tr> <td>Other</td> <td>15%</td> </tr> </tbody> </table>	Application	Relative Size (Estimated)	MRI	35%	CAT	25%	X-Ray	15%	EKG	10%	Other	15%	ID#27 / 1.15pm Copy from: <link #4>
Application	Relative Size (Estimated)													
MRI	35%													
CAT	25%													
X-Ray	15%													
EKG	10%													
Other	15%													

Fig. 7

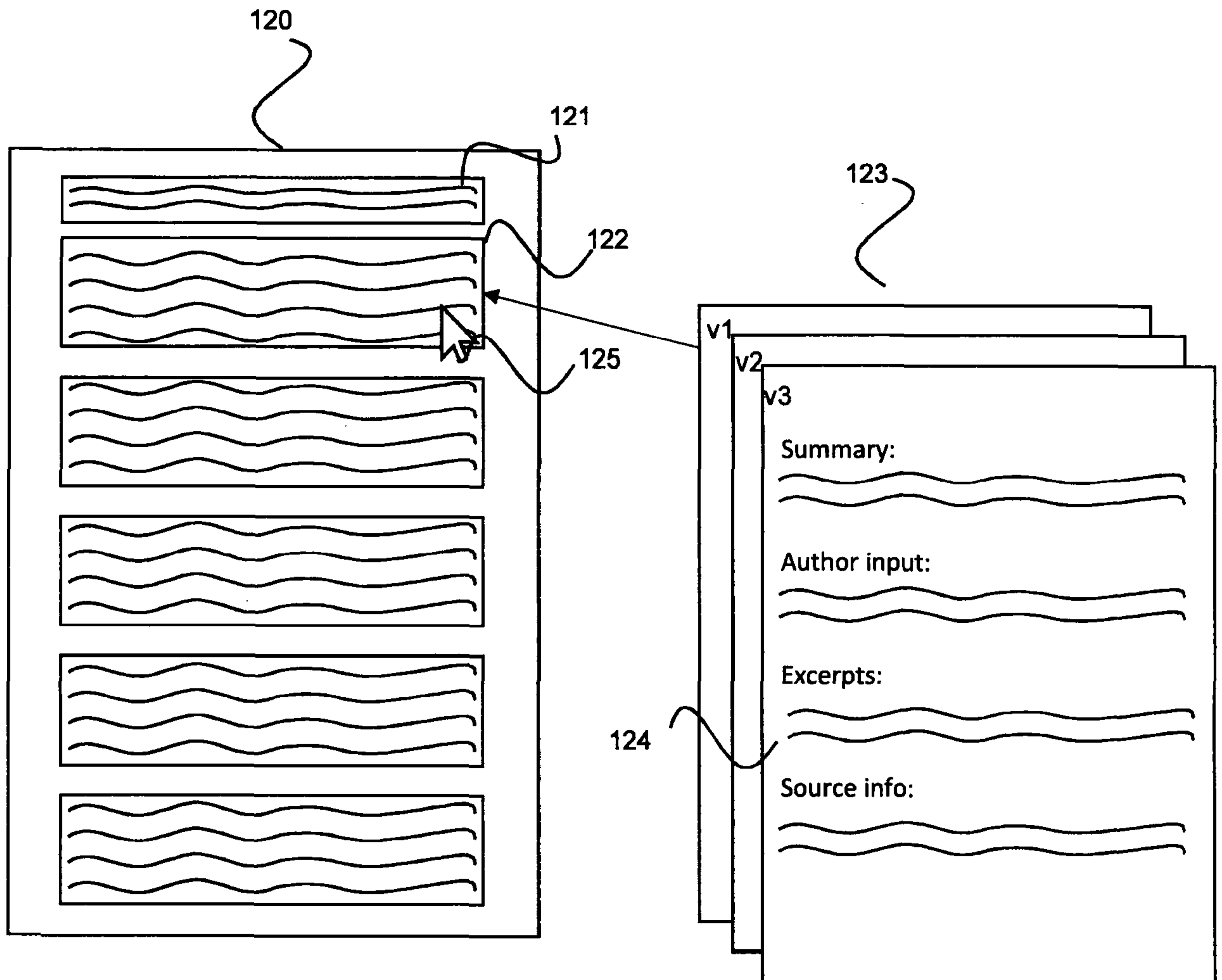


Fig. 8



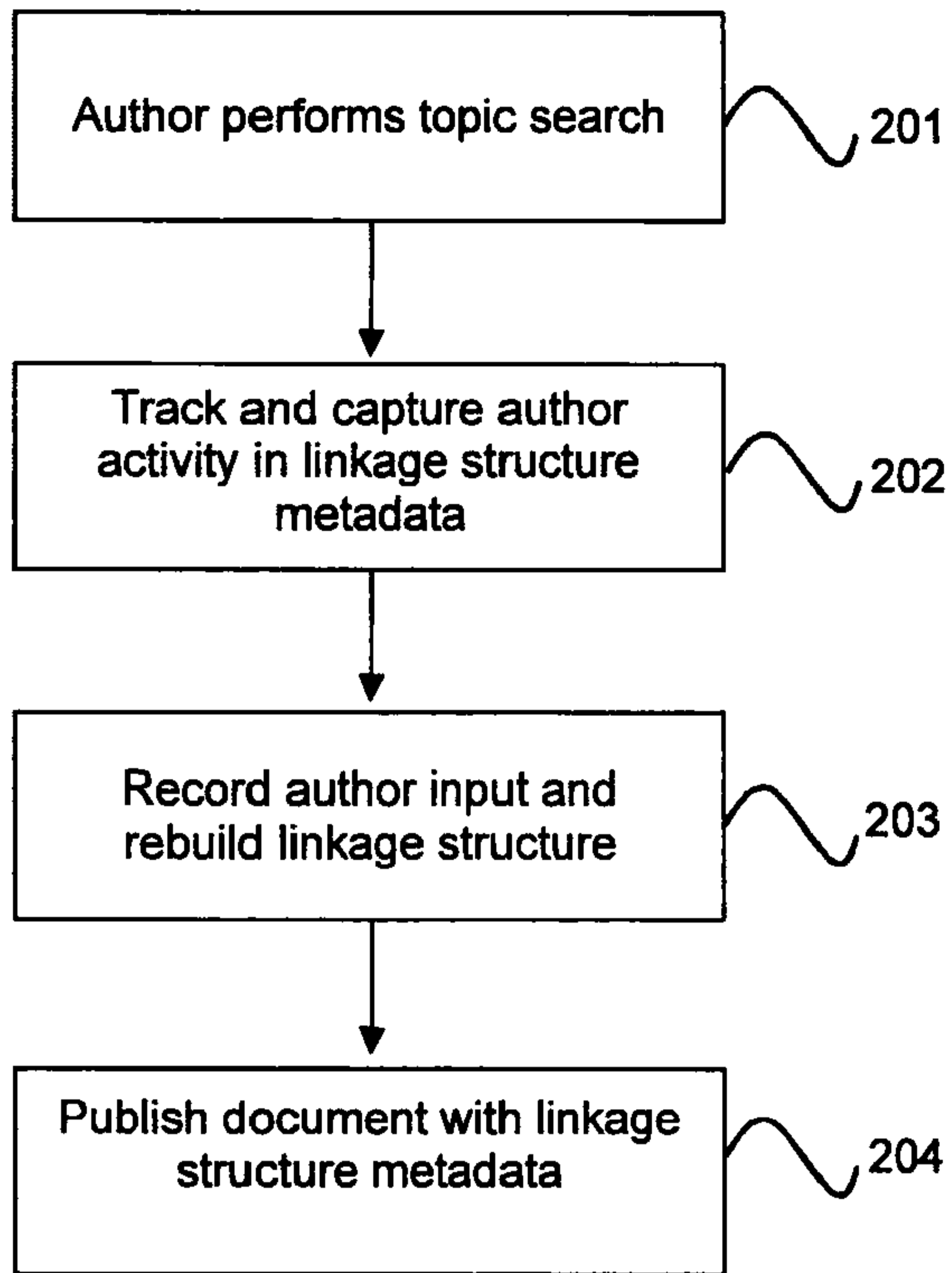


Fig. 9

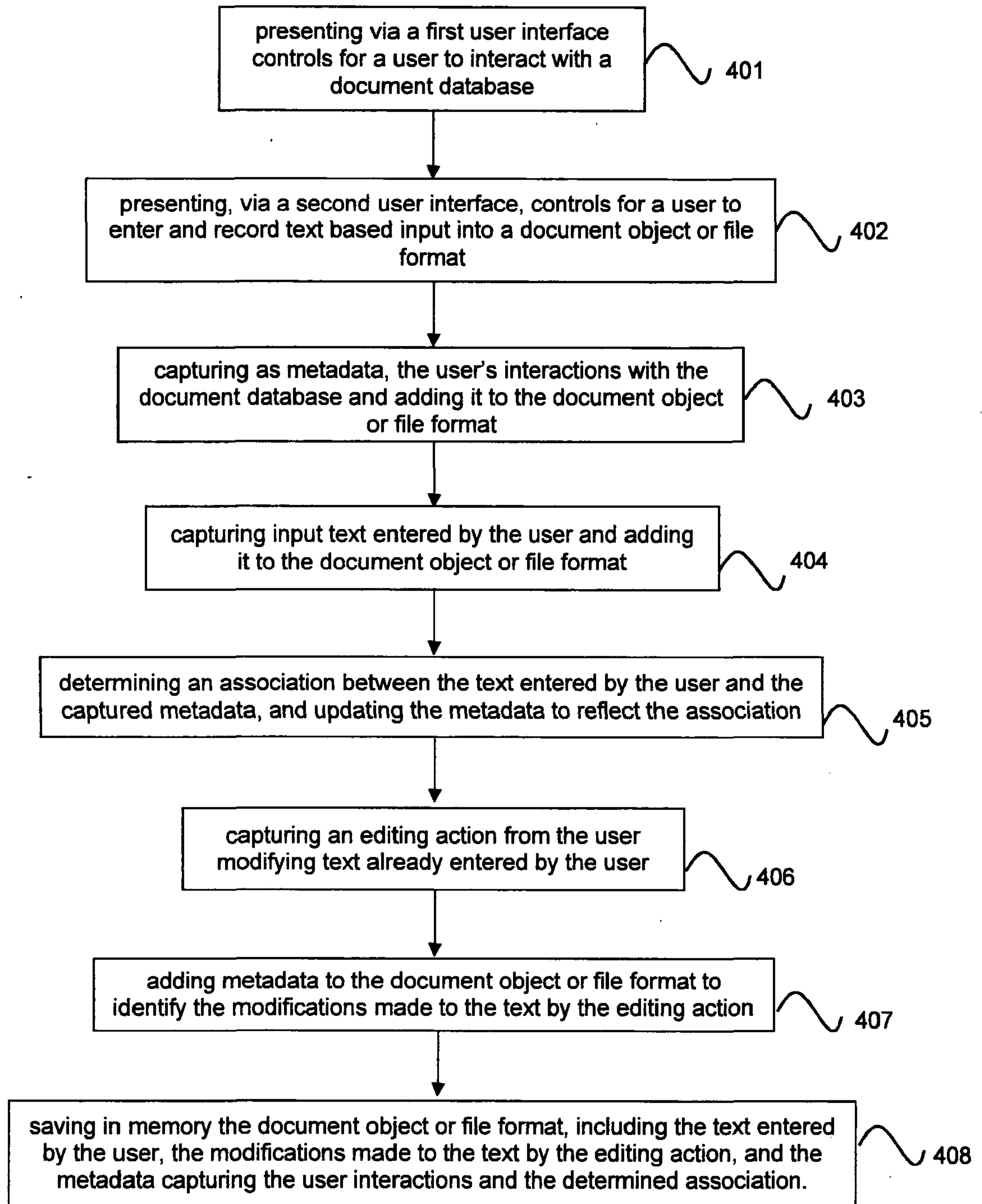


Fig. 10

# METHOD AND SYSTEM FOR CAPTURING METADATA IN A DOCUMENT OBJECT OR FILE FORMAT

## TECHNICAL FIELD

The application relates to a system and method for capturing metadata in a document object. The system and method also provide a corresponding file format in which the metadata and document data can be stored, and a user interface and system for end-to-end editing and publishing of text documents with the attached metadata.

## BACKGROUND ART

Conventionally, documents such as books, articles, reports and journals are written and read separately. During the process of writing a document, the author of the document may spend a great deal of time and effort researching topics and sources of information, choosing what subject matter to include and exclude. The final document eventually published by the author therefore offers only a static, polished view of the subject. If references or footnotes are provided by the author, it may be possible to find the factual basis for the author's views presented in the document. However, it will not be possible to determine what the author found during their research that they chose not to include in the final document. Further, it is not possible to determine what process the author followed on researching the subject, and preparing the document, and how the collation of the material in a particular order informed their final view.

Much information which may be valuable to a reader is therefore lost or discarded. Furthermore, once a document is published, it may be uploaded to a database such as the internet by the author. At this point, the integrity of the published document is lost, and the document may be read or modified by a third party without this being apparent to subsequent users. A recent example of the potential issues related to this is the widespread emergence of 'fake news', including reports and information that is not verifiable and often untruthful.

We have appreciated that it would be beneficial to provide a method and system for capturing information regarding the actions of an author during the process of writing a document, and providing this as metadata within a document object or file format. In this way, author contributions and interactions during the writing process, as well as key references and source information referred to or otherwise used in the writing process can be identified and maintained within the document object or file format, such that when a document is published, the reader is provided with an information-rich published document, and the integrity of the published document is verifiable.

35

## SUMMARY OF THE INVENTION

The present invention described herein relates to a system and method for capturing metadata in a document object or file format. The document object or file format may be used to track and record inputs to a document as metadata during the process of writing the document. In this case, the document object or file format forms an audit trail of metadata detailing the steps taken by an author or authors of the document from conception and research related to the preliminary writing of the document, through to reviewing, editing, and the eventual publication of the document.

The document object or file format may be used in an end-to-end authoring system for creating, editing and publishing documents. The authoring system is designed to analyse the sources used throughout the writing process and record them as metadata in the document object or file format. Once the document is written by the author or authors it may be published in the system. The system may be proprietary and secure, such that the editing of published documents is prohibited or tracked further using the document object or file format. Each published document is linked or published with the corresponding document object or file format for that particular document, providing the reader with an improved reading experience which is both more informative and more secure.

The file format of the present invention comprises a logical organisation of data in memory. According to the file format, a portion of the memory receives and stores data input by a user including one or more of text, numerical data and graphics such as images, tables and graphs. Another portion of the memory receives and stores data indicative of one or more characteristics of the user input data, the characteristics including at least one of: designation of the user who inputted the user input data such as a user identification; a date and/or time when the user input data was inputted; a source location such as a uniform resource locator (URL) address, indicating where the user obtained the user input data and/or what processes or information the user accessed before and at the time of inputting the user input data; an indication of the method of input the user performed to input the user input data; and any edits or modifications to the user input data, performed by a user, since the original user input the user input data.

The file format may be in the form of a table, directed acyclic graph, or any other structure in which user input data is associated, either by position or by an explicitly defined link, to the characteristics of the user input data. The file format is updated to create new entries each time new data is input or when existing data is modified by a user.

The file format is therefore configured to contain user input data as well as information concerning the input or modification of the data. This means that the file format of the present invention is very useful in tracking, as metadata, how a user inputs and modifies the content of a document during the writing process of the document.

The file format is configured to be accessible and readable when opened in a reader application configured to read the file format. This may be a proprietary system, hence keeping the contents of the file format secure outside of the reader application.

5

#### BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention will now be described, by way of illustration only, and with reference to the drawings, in which:

Figure 1 is a diagram of the system according to the present invention;

10 Figure 2 is a diagram of an example linkage structure according to the present invention;

Figures 3 to 7 are diagrams illustrating the structure of a logical file format for use with the present invention;

Figure 8 is an illustration of a published document according to the present invention;

Figure 9 is a flow diagram showing the method of the present invention; and

15 Figure 10 is a detailed flow diagram showing the method of the present invention.

#### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention relates to a file format or document object for storing data in a memory. The data comprises data input by a user, and metadata that describes characteristics relating to, amongst other things, how and when the user input data is inputted. In some 20 embodiments, the file format or document object is used to record user input data on a computer device and corresponding metadata for a document during the writing process of the document.

The file format describes the structure and the way in which data is stored in a 25 computer file. The computer file structured according to the file format of the present invention is saved to the memory of a computer readable medium. The memory may be implemented using any type of volatile or non-volatile memory devices, or a combination thereof, such as a static random access memory (SRAM), an electrically erasable programmable read-only memory (EEPROM), an erasable programmable read-only memory (EPROM), a 30 programmable read-only memory (PROM), a read-only memory (ROM), a magnetic memory, a flash memory, a magnetic or optical disk.

The computer device used for inputting user input data may comprise a processor, and various peripheral interface modules, the peripheral interface modules being, for example, a keyboard, a click wheel, buttons, and the like. The computer device may also include a power 35 supply component configured to implement power management on the device, a wired or wireless network interface configured to connect the device to a network, and an input/output interface. The device may operate based on an operating system such as Windows Server™,

Mac OS X™, Unix™, Linux™, FreeBSD™, or the like stored in the memory. The computer device itself may be a personal computer, tablet computer, mobile device such as an e-reader or a mobile telephone, wearable technology or the like.

5 The methods of the present invention relating to the file format, described herein, may be provided on a non-transitory computer readable storage medium including instructions, such as a memory including instructions, executable by the processor of the computer device. For example, the non-transitory computer-readable storage medium may be a ROM, a RAM, a CD-ROM, a magnetic tape, a floppy disc, an optical data storage device, and the so on.

10 The present invention will now be described in more detail, referring to the Figures where necessary. According to a first embodiment of the invention, a system for capturing metadata in a document object or the file format is discussed with reference to Figure 1.

Figure 1 illustrates a system 100 comprising a user interface module 101 and a data processing module 102. The user interface 101 comprises a text editing module 103 and a text publishing module 104. The data processing module 102 may be formed as part of a  
15 computer, server, or distributed computer network for example. The data processing 102 module includes or is communicatively coupled to a memory module 107. When data is recorded or saved to the data processing module 102, it is therefore saved to the memory module 107. The system 100 is communicatively coupled to a database 105. The user interface 101 and the data processing module 102 are configured to communicate with each  
20 other. For purposes of illustration, the database 105 may be the internet as shown in Figure 1, and accessed through a web browser. Content from the database 105 may include html data 105a. The web browser may include a database application 106, which is configured to communicate with the data processing module 102. Alternately, the database may be a proprietary database containing data available for use by the analyst or researcher, or a  
25 combination of a proprietary database and the Internet accessed through the web browser including the database application 106.

The system 100 provides an end-to-end platform for capturing metadata in a document object or file format, for a document in the process of being written. The system 100 tracks and records information that is used and accessed in the process of writing, and  
30 stores this information in a linkage structure. The linkage structure forms part of the file format or document object. This will be described later with respect to Figure 2 and a possible implementation of the linkage structure will be explained with reference to Figures 3 to 7. The linkage structure includes linkage metadata that describes not only where the written text of the document originates, but also how and if it is modified, and what actions  
35 are taken by an author during the process of writing. When writing process is complete, a document is published in the system 100, and the linkage structure or at least some of the linkage metadata is made available with the published document. The system 100 therefore

provides an information-rich document which not only includes text, but also the linkage metadata that defines how the text of the published document was found, inputted and modified. The combination of the text with the linkage structure means that users of the system 100 who read the published document have access to much more information than regular text documents. The document object or file format provided by the invention may be made available to the public, so that it is possible to determine for any document how it was constructed. In alternative embodiments, the document object or file format may be kept confidential within a proprietary system.

The document object or file format therefore provides attributions to the author or authors that are involved in the process of writing a document, starting from conception and research of the document through to the publication of the document. The document object or file format is configured to preserve, as metadata, information regarding the author interactions with the system 100 as well as information regarding sources that are deemed to be used by the author or are otherwise relevant to the document being written.

An advantage of this is that the reader, or other authors of the document, are provided with a network of information in the linkage structure in addition to the information of the published document. This additional information preserves the integrity of the published document, as it is clearly identifiable how information was ascertained and edited by authors.

The system 100 and the process of capturing metadata will now be described. The process starts by inputting data into the system 100. The text editing module 103 is configured to receive input data that is to be included in the published text document. Data is inputted into the text editing module 103 of the user interface 101 by a user 110 who performs one or more of three methods. The user 110 who inputs the data is known as the author. From here forth the input data is referred to as input text. However, it is to be understood that the data inputted by the author may include text data, numerical data, graphical data, or any combination of text, numerical and graphical data.

The first method of inputting text comprises directly inputting text into the text editing module 103. The author may therefore type, dictate, or otherwise input text into the text editor 103 of the user interface 101.

The second method of inputting text comprises pasting text into the text editing module 103. The pasted text is copied from a piece of text hereby referred to as a source.

The third method of inputting text comprises uploading all or a part of a predefined text document to the text editing module 103. The predefined text document may also be referred to as a source. This may be a PDF file or other graphical or image based file containing one or more of text or graphical information.

The text-editing module may include a predetermined template in which the author inputs text into different sections. When input text is received via one of the first second and/or

third methods at the text editing module 103, the text editing module 103 communicates with the data processing module 102 to determine whether any linkage metadata exists relating to the input text. The linkage metadata contains the source or sources used by the author as well as other related tracking data concerning the author of the input text.

5 Usually, the author performs a search of the database 105 for sources relevant to the intended topic of the text document before inputting text. The database 105 may be any structure capable of storing information that is retrievable by the system 100, such as the internet accessed through a web browser as seen in Figure 1, or a propriety database as discussed previously. When the author conducts the search, the data processing module 102  
10 tracks the actions of the author, the actions including any interactions with the database such as clicks, website visits and downloads for example. The actions of the author may be recorded in the linkage structure by the data processing module 102.

When the author uses the internet as the database 105 to search for information from sources, the recording of author actions and the related data may be facilitated by the  
15 database application 106, which provides the communicative coupling between the database 105 and the data processing module 102. The database application is a web-browser extension, plugin or the like.

In the second and third methods for inputting text above, the sources identified in the search are directly used for inputting text into the text editing module 103, by copying and  
20 pasting from a source or by uploading a source document respectively. However a source may still be indirectly in the first method for inputting text, in that the author may obtain information from a source found in the search, and use said information for informing the text which they input into the text editing module 103. Therefore, a source may be used for each of the three methods for inputting text into the text editing module 103.

25 Once a source is identified in the search or used for inputting text, it is recorded by the data processing module 102 in the linkage structure. The linkage structure contains linkage metadata, which describes how the source is used to inform the text inputted into the text editing module 103 of the user interface 101. The linkage structure maps each source used by the author from the database 105 to the corresponding inputted text. When the author  
30 inputs separate pieces of text from different sources into the text editing module 103, using any of the possible three methods for inputting text, the linkage structure maps each separate piece of inputted text to its respective sources.

For instance, if the author visits a website, the visit of the website will be captured by in the linkage structure as an author action. This includes recording the Uniform Resource  
35 Locator (URL) address of the visited website. Data related to the website itself may also be recorded as a source, such as the html and text content of the visited website. This may then



be stored in the linkage structure as the source corresponding to the user action of visiting the website.

Sources are recorded by the linkage structure either at the time they are identified in the search or when the author inputs text, or an image for example, that relates to a particular source.

When the author visits a webpage for example, the html data and other text content from the webpage may therefore be extracted and recorded in the linkage structure at the data processing module 102 at the time it is identified in the search. Alternatively, once the data processing module 102 receives an indication that the author has inputted text, the data processing module 102 may communicate with the database 105 to search for a particular source relating to the inputted text, according to the linkage metadata of the linkage structure that corresponds to the captured author actions. The search of the database 105 may be done only according to linkage metadata that is first determined to be relevant to the inputted text. For instance, if the user inputs text concerning the topic of 'the economy', the text editing module 103 may communicate the topic of 'the economy' with the data processing module 102, such that the data processing module can identify any linkage metadata relating to the topic. The data processing module 102 may then communicate with the database, through the database application 106 or otherwise, to identify any sources relating to the linkage metadata concerning the topic of 'the economy'. If sources, such as a website previously visited by the author, are found in this way they are then recorded to the data processing module 102. This alternative method may be considered more efficient because source information is only recorded to the data processing module 102 once it is determined that it has been used to inform the inputted text.

Some examples of how sources may be recorded in the linkage structure according to the first, second and third method for inputting text are provided here.

For the first method of inputting text, the inputted text may be compared, semantically or otherwise, to source information recorded in the data processing module 102. If the comparison determines that there is a match between a portion of the inputted text and the source information, then the linkage structure is updated with linkage metadata to reflect that the portion of inputted text was informed from the specific source information. It is to be understood that a 'match' does not necessarily mean that text from the source has been directly copied in the inputted text. Alternatively, a match may occur if there are semantically similar words, sentences or statistics in the inputted text as in the text from the source. Determining that there is a match between the inputted text and the text from the source may be subject to a similarity threshold score, above which the inputted text is identified as a match of the text from the source.

The linkage metadata may also reflect who the author was that inputted the text, what changes, if any, they made to the original source information, and when the text was inputted compared to when the source information was accessed.

5 It is to be understood that the comparison between the inputted text and the source information may be performed according to any text or language based recognition algorithm, examples of which will not be discussed here.

10 For the second method of inputting text, when source information is copied from the database 105, the database application 106 tracks and captures the copy action of the author, and sends the copied text to the data processing module 102. The copied text is sent to the data processing module 102 with linkage metadata that includes the URL related to the copied text, for example. The linkage metadata is used to update or build the linkage structure that exists in the data processing module 102.

15 When the author pastes the copied text into the text editor module 103, to effectively input the copied text, the text editor module 103 then communicates with the data processing module 102 to indicate that text has been pasted. The data processing module 102 then identifies that the text inputted in the text editor module 103 is the copied text, and is thus able to attribute the inputted text to the source information and associated linkage metadata received from the database application 106 when the original source information was copied.

20 The second method of inputting text therefore involves recording the copied text and associated linkage metadata from the database 105 once it is copied, such that if the text is pasted into the text editing module 103, the data processing module 102 already contains the recorded source information and linkage metadata necessary for linking the source information to the text inputted by the author. Alternatively, as mentioned previously, when the text from the source is originally copied, the author action of copying the source information, including the linkage metadata, may be sent to the data processing module 102 from the database application 106 without the source information itself. In this alternative embodiment, the source information, such as html data, is only retrieved from the database 105 once the copied text is pasted into the text editing module 103. This alternative method may therefore be considered to be more efficient as source information is only recorded in the data processing module 102 once input text is pasted. The text editing module 103 then communicates with the data processing module 102 and the data processing module 102 communicates with the database 105 via the database application 106 to obtain the specific source information.

35 For the third method of inputting text, the uploaded document itself may be recorded as an online source by the database application 106 and sent to the data processing module 102.

The third method of inputting text may however occur offline. In this case, the database 105 may be contained within the system 100. The author may have already saved the predefined document, or the database 105 itself may be proprietary as discussed previously.

5 In either the case where the database 105 is accessed offline or online, when a predefined document is downloaded, the database 105, or the database application 106, communicate with the data processing module 102 such that linkage metadata of the downloaded document is captured and used to update the linkage structure. Furthermore, the document may be stored as a source at the data processing module 102.

10 It is to be understood that the author may also input text including numerical data into the system 100 as discussed previously, according to any of the first, second or third methods for inputting text. The text including the numeric data is considered as text for the purposes of tracking and recording metadata in the linkage structure of the document object or file format as discussed above.

15 Optionally, the system 100 may also be configured to provide input text to the text editing module 103 according to a fourth method, which focuses on the retrieval and manipulation of numerical data in a source. This fourth method for including input text comprises, firstly, identifying, by the data processing module 102, numerical data in a source within the database 105. The source may be a source identified in the initial search performed by the author, or otherwise identified by metadata in the linkage structure of the document  
20 object or file format. The system 100 may alternatively search the database 105 in the background for numerical data relating to the subject matter of input text already received by the text editing module 103. In order to achieve this, the system 100 may perform text and language recognition on the input text acquired from the author.

25 The numerical data may be part of a graph or table contained in a source. In this case, the graph or table is a structured data set. Graphs or tables that have structured data sets comprise formatting, such that values within the graphs or tables are fixed in length, font, or relative position for example. The graphs and tables may have titles, subheadings or axis titles that can be identified by the system 100 to determine the subject of certain numerical values. For instance, if a source comprises a table with the column title 'quarterly revenue', the system  
30 100 identifies that the numerical contents of that column are associated with the quarterly revenue. The linkage structure may be updated with linkage metadata relating to the numerical data from the structured data set and the corresponding relationship to the source from where it was found. The system 100 may also identify unstructured numerical data, such as numerical values found within extracts of prose. In this instance, the system 100 may  
35 identify the numerical value using a text-based recognition algorithm. The system 100 may then identify a relationship for the numerical value with a specific subject, by performing language recognition on the sentence from where the numerical value was identified. For

instance, for the sentence: *'The quarterly revenue for the third quarter was £2,345,800'* the system 100 may first identify the numerical value '£2,345,800' using text recognition, before using language recognition to determine that the numerical value is associated with the third quarter revenue. The linkage structure may then be updated with linkage metadata relating to the numerical data and the corresponding relationship to the source from where it was found.

The fourth method for inputting numerical data into the text editing module 103 may then include aggregating and combining numerical values that are related by a certain subject. This may involve the system 100 identifying common subjects in the linkage structure according to the linkage metadata of the numerical data. For instance, numerical values that relate to 'quarterly revenue' are combined by the system 100. The system 100 may then perform an analysis and processing on the combination of numerical values, such as applying mathematical formulae, or segregating related values dependent on sub-divisions of the subject such as the year or date, for example, in order to produce a meaningful result. The result may then be used to produce a graphic such as table or graph. The graphic is then provided to the text editing module as a graph or table to be published. The graph or table to be published contains numerical data that may be accumulated from several sources, but that is interrelated according to the specific subject. The table may for instance, show the quarterly revenue for a company searched or written about by the author elsewhere in the writing process. In this respect, the system 100 may perform a search of the linkage structure and/or the database 105 for numerical data according to sources author by the user or input text inputted by the author. The manipulation and analysis may occur in the background and be presented to the author in the text editing module 103 as a finished graph or table to be published. The author may then be able to accept, reject or edit the table or graph for publishing. The linkage structure of the document object or file format is updated with metadata according to the table or graph for publishing and the author's interactions with it, in terms of accept, reject or edits. In this way, the document object or file format preserves the source information and manipulation of the numerical data, such that when a document published in the system 100 contains graphs or tables, the document object or file format provides the reader with information regarding the source, references, and any mathematical manipulation of the original numerical data.

It is to be understood that the method of inputting numerical data according to the fourth method is different from the first, second and third method for inputting text in that the system 100 automatically performs the input in the fourth method, by using text and language recognition, followed by data analysis and processing.

Once the author has finished inputting text into the text editing module 103 in the user interface 101, the publisher module 104 is configured to publish the text document that contains the inputted text. The publisher module 104 publishes the text document in the user

interface module 101. The published text document includes the at least some of the linkage metadata of the linkage structure used to map the use of sources from the database 105 to the inputted text.

5 A user of the user interface module 101 may access the published text document through interaction with the user interface module 101. It is to be understood that the user of the interface module 101 who accesses the published text document may or may not be the author of the published text document. Therefore, the user who accesses the published text document is referred to as a 'reader'.

10 Since the published text document contains the text input of the author as well as at least some of the linkage metadata that describes how a source or sources were used to inform the input text, the reader is provided with an information-rich publication, from which sources of further information can be easily identified.

15 This allows the reader to discern whether the author has used reputable sources, whether the author had edited information from any sources, or simply whether the author has used a reliably large number of sources in forming the published text document. Since the originally inputted text and the published text in the text document are contained in the same system 100, and the linkage structure containing linkage metadata is also contained in the system 100, the system 100 promotes reliability in terms of authorship and authenticity. These are only a few examples of the advantages of the present invention.

20 Figure 2 shows an example of the linkage structure 110. The linkage structure 110 maps how a source 111 and an author input 112 into the text editing module 103 are used by the system 100 to arrive at the published document 113. The linkage structure 110 also includes linkage metadata 114 that provides information on author actions, such as a click, scroll, website visit, copy or the like, and how these actions link between sources 111 and author inputs 112. The linkage structure 110 forms at least part of the document object or file format of the present invention.

30 In one embodiment, as shown in Figure 2, the linkage structure 110 is arranged in a Directed Acyclic Graph format (DAG), wherein the nodes of the graph can only be traversed in one direction. In Figure 3, the linkage structure 110 contains multiple sources 111 and author inputs 112, and maps the entire creation of the published document. However, it is to be understood that the linkage structure 110 may overwrite itself if a second author input overwrites input text of a first author input. An overwrite of the linkage structure may be performed to save space on the data processing module 102.

35 The processes of building and/or updating the linkage structure 110 will now be discussed.

Initially, as the author searches for sources of information on the database 105, the data processing module 102 tracks the actions of the author as discussed previously, the

actions including any interactions with the database such as clicks, website visits and downloads for example. The author actions 114 are recorded within the linkage structure 110. The author actions 114 stored in the linkage structure 110 may be directly related to sources 111. For instance, the author actions 114 may be a visit of a particular website, and the source 111 may be the text or html content of the particular website. The source 111 in this instance may be incorporated into the linkage structure 110 at the time of the search done by the author, or it may be included in the linkage structure 110 at a later time, once the author has inputted text into the text editing module 103 that is determined to be informed by a particular source as discussed previously. It is to be understood that incorporating the source 111 into the linkage structure 110 may comprise including the whole source, including extracted text and/or html data, or alternatively, it may comprise including a pointer or address to the source, so that the source can be accessed only when necessary and without putting undue burden on memory allocated for the linkage structure 110, for example.

When the author inputs text into the text-editing module 103, the input is recorded and sent to the data processing module 102, where it may be added to the linkage structure 110. At this point the linkage structure 110 contains metadata corresponding to the source 111, the author actions 114 and the author input 112. The author input 112 is compared to the source 111 to determine in what way the author has used the source 111 to inform the text input into the text editing module 103. In other words, the method of text input, from one or more of the first, second and third method of inputting text is determined and the associated author action, such as typing, copy and pasting, or uploading is recorded in the linkage structure 110. According to the first method of inputting text, there may be the further step of directly comparing inputting text with the source text and scoring a similarity to determine whether the source text was used to inform the input text, as discussed previously.

The linkage structure 110 thus contains information regarding what sources are used for input text, and where and how they are used in the input text. As can be seen in Figure 3, two or more sources may be used for one input text, such that words, sentences or paragraphs from different sources can be inputted into the same input text and still be tracked and linked back to their respective sources 111 by the linkage structure 110.

Furthermore, in an optional embodiment, the linkage structure 110 may keep track of inputs from separate authors, who collaborate on writing into the word editing module 103. The first author input 'AUTHOR INPUT#1' 112 and the second author input 'AUTHOR INPUT#2' 112 in Figure 3 may therefore represent the inputs of different authors. The example of Figure 3 shows that the second author input uses the first author input, but also adds from the source 111 used by the first author as well as from a new second source 111. The actions that the second author performs on the first source, the second source and the first author input are recorded as author action metadata 114.

The linkage structure 110 is therefore capable of keeping track of revisions and edits to the inputted text by separate authors, and how each author input interacts with source information and other author inputs.

5 In an alternative embodiment, to save space in memory module 107 coupled or included in the data processing module 102, the linkage structure overwrites linkage metadata such as source information 111, author actions 114 and author inputs 112 if a previous author input 112 is overwritten by a later author input 112.

10 When the author or authors have finished inputting text into the text editing module 103, the text editing module 103 communicates with the data processing module 102 to indicate that text inputting has ended. The linkage structure 110 updates accordingly to reflect that the inputted text is used in the published document 113. Figure 3 therefore shows an example of a complete linkage structure 110, from the sources 111 used to the finished document 113.

15 The use of the data linkage structure and its implementation in a document object or document file format will now be described by way of example with reference to Figures 3 to 7.

20 Figure 3 shows a data structure for capturing metadata according to an embodiment of the invention. A tabular structure is used for the purposes of illustration and to show how the different aspects of the document object data may be associated with each other. In the example illustrated, the document object has three columns for storing respectively a data linkage item number 301, a data linkage content item 302 and a metadata linkage object 303. The data linkage item number 301 provides a unique ID for each row in the table so that it can be referenced unambiguously. The data linkage content item 302 contains an element of text or a graphics element input by the user, using any of the techniques noted above, and metadata linkage object 303 contains metadata detailing when and how the content item 302 was created.

30 Figure 3 shows the case where the user has begun a new document or a new section of a document and entered text into the text editor. The user is creating a document about "Artificial Intelligence in the healthcare industry" and has begun by typing the words "*The opportunities for Artificial Intelligence in health care*" into the text editing module 103 via the user interface 101. The "/r" symbol indicates a carriage return.

35 The data processing module 102 detects the text entry and populates the data object model 300 with the text data that the user has entered, the carriage return information, as well as metadata, such as the identity of the user entering the text, the time the text was entered by the user (12.56pm), and the nature of the input – in this case "manual input". The user is represented by an indicator ID#27 by way of example. More or less metadata may be captured in other implementations of the preferred embodiment.

The user then browses through the database 105 to discover documents that relate to this topic. As noted above the database 105 may be the Internet, or a proprietary database in which reports or other documents are made available. During the user's search for relevant sources, the user finds and view a number of documents with respective individual links <link #1>, <link #2>, and <link #3>. At 1.02pm, the user cuts and pastes the following paragraph of text from the document available at <link #1> into the text editing module 103:

- *Artificial Intelligence provides many opportunities for improving healthcare, particularly with application to medical diagnostics. Medical staff typically diagnose a patient's underlying health issues, based on a visual inspection of a patient medical data, such as MRI or EKG data. Diagnosis is based on the ability of the medical professional, their experience, and sometimes luck in spotting the tell-tale signs of disease. Some conditions may be more difficult to detect than others.*

The data processing module 102 detects that the user has both browsed to the document in the web browser or a proprietary search window provided in the database module 105, and copied and pasted text from the document into the text editing module 103. The text copy and pasted by the user appears in the text editing window underneath the text that the user typed and which is stored in record #1. Thus, the data processing module captures the data as new record #2. Metadata is stored with the input text to indicate that the cutting and pasting operation was carried out at 1.02pm, by user with ID#27. The entry type is therefore "copy from" and the link from which the text was sourced is also added to the metadata, <link#1>.

At 1.05pm, the user also copies and pastes the following paragraph of text from the document available at <link #2> into the text editing module 103:

- *Machine Learning algorithms, such as those provided in the field of Artificial Intelligence, can also be trained using real medical data, benefitting from the collective experience of the medical professionals involved in the training. Such algorithms can typically extrapolate that training better than humans, reducing the element of luck in the diagnosis. Machine learning algorithms provided by Medical AI Ltd have shown to be 20% more reliable than doctors in actual field trials.*

In a similar fashion, the data processing module 102 detects that the user has both browsed to the document in the web browser or a proprietary search window provided in the database module 105, and copied and pasted text from the document into the text editing module 103. The text copy and pasted by the user appears in the text editing window



underneath the text that the user typed and which is stored in record #2. Thus, the data processing module captures the data as new record #3. Metadata is stored with the input text to indicate that the cutting and pasting operation was carried out at 1.05pm, by user with ID#27. The entry type is therefore “copy from” and the link from which the text was sourced is also added to the metadata, <link#2>.

In this way, the record numbers will be understood to reflect the spatial arrangement or layout of the text in the document.

During the user’s search, the user browses to and reads a document at <link#3>. However, the user does not copy and paste any text from this document into the text editing module 103. The data processing module 102 captures this fact by creating a record #4 that stores the actions of the user, <link#3>, “1.06pm browsed”. If the user subsequently cuts and pastes text from the third document, available at <link#3> then record #4 may be updated in the manner illustrated in records #2, and #3 with a new time and a “copy from” indication above the link.

If the user never copies and pastes text from the link <link#3>, then record #4 remains in the metadata as a record of the user’s actions browsing and inspecting the third document, that is available for later auditing. In this case, the position of the record #4 is not indicative of the layout or special arrangement of the text information in the document model, and the data processing module 102 attempts to keep the record #4 positioned in the document model as close to the correct time as possible according to its time stamp. This will be illustrated in Figure 7. In this context, the fact that the user does not copy and paste text from the third document can be detected if the user browses away from the third document to another document, or begins another action, such as manually typing text into the text editor.

In the event that the user does not copy and paste text from the third document, but does begin typing into the text editing module 103, embodiments of the invention may detect whether the text being typed by the user matches the text present in the database 105 document window or web browser window, so as to capture the fact that the user is copying the text, but has not used the clipboard function.

In Figure 5, once the user has input a number of paragraphs of text into the text editing module by different methods, the user begins to edit the text in the document model. This is carried out in the usual way, using the user interface provided by the system 100 and the text editing module 103. As shown in Figure 4, the user takes the view that the first paragraph copied from the first document is overly long, and at 1.07pm trims the section of text to say:

- *Artificial Intelligence provides many opportunities for improving medical diagnostics, in which medical staff typically diagnose a patient's underlying health issues, based on a visual inspection of a patient medical data. Diagnosis is based on the ability of the*

*medical professional, their experience, and sometimes luck in spotting the tell-tale signs of disease. Some conditions may be more difficult to detect than others.*

The words “healthcare, particular with application to”, “such as MRI or EKG data” have been deleted, and the first and second sentences have been run together by the phrase “in which”.

The data processing module captures the actions of the user and updates the document model accordingly. Sections of the original copied text that are unedited by the user are preserved as new separate records #2 and #4, and a new record #3 is created reflecting the edits made to the original text. The records with a higher number are renumbered accordingly. The metadata for each of the data linkage content items #2 and #4 is updated to “unedited”, while the metadata for record #3 is set as “edited”, and data showing the edits is preserved. In this way, the document model records both the fact that the user’s document contains text copied from another document, and the fact that this text has been changed. Using the record structure of the document model, it is possible to determine what aspects of the copied text have been edited and what aspects have not been edited.

In this example, record #3 is created to reflect the editing of a portion of text beginning with the user’s first edit and ending with the last. It is possible to make the creation of records and the indication of what sections of the text are “edited” or “unedited” more or less fine-grained. For example, records marked “edited” might be limited to show only adjacent deletions and insertions of text, such that in Figure 5, the section of text in record #3 that reads “*staff typically diagnose a patient’s underlying health issues, based on a visual inspection of a patient medical*” would be included in a new record marked “unedited”. However, to avoid proliferation of records (and to make the description here easier to follow), it will be understood that record #3 strictly shows both edited and unedited text, and that record #3 shows a section of edited text having both edited and unedited sections.

In example embodiments, the data processing module 102 may make the decision of whether to create a more fine grained record, or a record showing a section of edited text, based on the numbers of words in the record to be marked as a section of “edited” text. For example, in Figure 5, if record #3 contained more than 250 characters for example, data processing module 102 might seek to create more than one record to capture the associated editing information.

Figure 6, shows a further editing action performed by the user and captured by the data processing module 102. At 1.08pm the user deletes the final sentence of text “*Some conditions may be more difficult to detect than others*”, and the data processing system captures this deletion as a further edit of the text from link <link#1> creating new record #5, and renumbering former records #5 and #6 to be #6 and #7.

As discussed above, a further method of data or text entry might be to upload data such as an image of text data, a chart, or some other graphical image. Most computer systems offer functions for capturing as a graphic part of the screen, or saving a link which may then be pasted or uploaded into a document and reproduced. In this example, the user finds at 5 1.15pm a chart which illustrates the applications of image based processing techniques to different types of patient data, and has added this chart to the document being created.

The data processing module 102 captures the link information (a shortcut or pathway to the file) or the graphical information (the image file itself) of the chart and includes this in the document object model as the data linkage content item. As before, metadata is added to 10 the document model to indicate the user ID, the link where the chart was found, <link #4>, and the time of addition.

In this way, as the user creates and edits the document, the system tracks the user activities and captures in the document object model, the necessary data describing the content and provenance of the data. It is to be understood that the data may refer to text data 15 or numerical data, or both.

When the document is published by the author in the system 100, the linkage structure captured in the document object model or at least some of the linkage metadata may be made available to the reader. This feature will now be discussed with reference to Figure 8.

Figure 8 shows an example of a published document 120 according to the present 20 invention. The published document 120 includes a title 121 and a body of text. In the body of text, a portion 122, such as a paragraph, may be selected by a reader by a click, hover or the like 125. Selecting the portion of text 122 opens a textual representation 123 of the linkage structure 110. The textual representation of the linkage structure 123 includes linkage metadata 124 related to the selected portion of text 122.

The published document 120 is therefore information-rich, in that the reader of the 25 published document can interact with the document 120 to obtain more information 123 regarding the document 120. The metadata 124 included in the textual representation 123 of the linkage structure 110 is determined based on the selected portion 122 of the published text document 120. As shown in Figure 8, the metadata available may include a summary of 30 the selected portion of text 122, what parts of the text are inputted by the author, what parts are excerpts or quotes direct from sources, and information regarding the sources. Furthermore, as can be seen from Figure 8, the textual representation 123 may also include revision or tracked history data, showing version 1 v1, version 2 v2, and version 3 v3 of the selected portion of text 122.

The linkage metadata information for each portion of text 122 is retrieved from the 35 linkage structure 110 from the data processing module 102 once the text inputting is complete. The publishing module 104 then publishes the inputted text with links to the relevant linkage

metadata as shown in Figure 8. When showing the source metadata information, the textual representation 123 may include an address or link to an entire source document, or may alternatively highlight and include relevant parts of the source document.

5 An advantage of the present invention is that the reader is able to quickly and efficiently have access to much more information regarding the topic of the published document 120 than is actually in the document itself, by simply having access to the system 100. Since the document is written and published in the system 100, and the system 100 maintains all linkage metadata information and the linkage structure 110, when the reader enters the system 100 they may have access to all recorded and saved information.

10 It is to be understood that the system 100 may include several published documents 120, each with their own respective linkage structures 110. In this way, the system 100 becomes a database in itself, in which authors can contribute to new or existing published documents, and readers can read the documents 120 as well as see exactly what authors have done and what sources they have used according to the linkage metadata.

15 Each linkage textual representation 123 of a linkage structure 110 may further link to sources identified in similar linkage structures from different published documents 120.

In other words, once there are several published documents 120, historic data relating to the different published documents 120 may be used and provided by the different respective linkage structures 110. Furthermore, when the author performs a search for a topic according to a new document, the system 100 may first search for similar published documents 120 and their corresponding linkage structures 110 for historic data that is relevant to the topic of search, and suggest them to the author. The search for similar published documents may be done using language recognition algorithms for example.

20 The linkage structure 110 of each published document 120 is maintained such that the reader and/or author can audit the writing process and determine the reliability of publications. Since the linkage structure 110 is maintained, this promotes security within the system 100 and clarity of information to the reader.

25 Furthermore, the system 100 may have a feedback functionality. The feedback functionality is provided using the data collected by the database application 106 and stored in the data processing module 102. In particular, the database application 106 captures sources and related information as discussed previously, which is then recorded on the data processing module 102. The database application 106 may also track the navigation behaviour of the author on the database, such that website visits, search parameters, and number of clicks, for example, are recorded by the database application 106. This information may then be analysed to determine search practices of the author and how sources were used during the report writing process. This includes identifying what search terms resulted in finding a source in the search which was then used by the author, which sources were visited

and used, and which sources were visited but not used by the author. This analysis of the search and the usefulness of sources may be used to inform the author or an administrator of the system 100, via a metric such as a score, search analysis document or graphic, of the productivity of the author and/or the quality of the search and output of the search. Essentially  
5 this means that the author or administrator of the system can determine author related statistics concerning the report writing process, such as how productive and efficient the author is in searching and identifying relevant source information, as well as where to find relevant source information. Using this feedback allows the administrator to see which authors create specific reports, and what behaviour in terms of the search led to good or bad quality  
10 output or to fast or slow delivery of the report.

In a further embodiment, it is also possible that the system 100 automatically produces these author-related statistics, and report on the results. The report of the results may take the form of a score, search report document or graphic, such as authoring time vs search behaviour, or report feedback scores vs search behaviour.

15 The author-related statistics may then be published to identify how an author is performing, and which areas or addresses of the database, and therefore which sources, are most used and most useful. In this way the sources used in previous report-writing may be ranked from most used to least used, or according to most commonly found during a search. This ranking may be published within the system 100 for the administrator and one or more  
20 author to view. Furthermore, during a subsequent report-writing process, when an author navigates to a source previously visited by an author of a previous report, the database application 106 may prompt the author with author-related statistics regarding the source. For example, the database application may display information to the current author relating to the interactions of the previous author or authors with the source. This may include stating that  
25 the previous author or authors used the source or did not use the source, and statistics and/or rankings regarding how popular the source is with authors, or common search terms inputted by authors who used the source.

This feedback functionality may also apply to specific paragraphs within sources, such that paragraphs within sources are ranked according to their selection or non-selection by  
30 previous authors. Manually selected paragraphs may therefore also be included in the author-related statistics.

The aim of the feedback functionality is to drive ongoing improvements to the report-writing process, so that the system administrator and/or future authors are informed with historic data concerning previous report-writing processes.

35 It is to be understood that although the published document 120 is accessed in the system 100, it may be exported and read by the reader outside the system 100. However, in

order to maintain security and to keep the linkage structure updated, it is preferable that the author or authors of a document perform the process of writing within the system 100.

In the above description, the system 100 is referred to as a collection of modules and a database 105. It is however to be understood that the system 100 may be comprised by any  
5 computer terminal or a set of computer readable instructions recorded on a computer-readable medium. The system 100 may therefore exist over a network, server or the like and as such is not required to be accessed locally. The author and/or reader may access the system 100 over a connection to the internet or any other remote connection to the system 100.

The method according to the present invention will now be described. It is to be  
10 understood that the method of the present invention relates directly to the workings of the invention as discussed above with reference to Figures 1 to 4. Like the system 100 of the present invention according to these Figures, the method according to the present invention may be performed on a set of computer readable instructions recorded on a computer readable medium, such as a computer program written on a computer, sever, or distributed  
15 computer network.

The method according to the present invention is illustrated in Figure 9.

In step 201 of the method according to Figure 9, an author of the document to be written may perform a first search according to the topic of the document to be written. The author searches for sources comprised within a database. During the search, the author may be prompted with  
20 sources or previously published documents relating to the topic of the document to be written, from searching historical data. Similarly, the author may be provided with a predetermined template for inputting text according to the topic of the document to be written. The topic of the document to be written can be determined using language recognition algorithms or other text-based analysis on the search string of the first search.

In step 202 of the method according to Figure 9, the activity of the author is tracked  
25 and captured and used to build a linkage structure including linkage metadata. The activity of the author that is tracked and captured includes any interaction from the author during the first search, and with sources of information identified in the search, or sources otherwise found by the author. These interactions may include clicks, website visits, downloads and copy and  
30 pastes for example. Throttling may be performed on the tracked actions of the author, such that only significant actions that are useful in determining sources are captured and tracked. At this stage of the method, sources may be identified according to the author actions and added to the linkage structure. Such sources may include websites, spreadsheets, or other such documents. The linkage structure connects the sources to the related author actions. For  
35 example, where the author action is a website visit, the source connected to that user action may be the html content of the website.

In step 203 of the method according to the present invention, author input is recorded at a user interface, the author input being an input of text. The author may input text to the user interface according to any one or more of three methods.

5 The first method of inputting text comprises directly inputting text into the user interface. The author may type, dictate, or otherwise input text in this way.

The second method of inputting text comprises pasting text. The pasted text is copied from a piece of text originally found in a source.

The third method of inputting text comprises uploading all or a part of a predefined text document.

10 When the author inputs text using any one or more of these methods, the linkage structure is updated according to the author input, and if the source used is not already recorded, it is then fetched and recorded in the linkage structure.

15 When the author inputs text according to the first method of inputting text, the linkage structure is searched for sources that compare closely with the inputted text to determine if the author used these sources in informing the inputted text.

When the author inputs the text according to the second method of inputting text, the linkage structure is updated to reflect that the author has copied and pasted from a particular source. Linkage metadata from the source, such as a URL, html or text data is included in the linkage structure.

20 When the author inputs text according to the third method of inputting text, the linkage structure is updated with linkage metadata signalling what downloaded source was uploaded as input text, and if necessary what portion of the source was used. Step 203 repeats every time that new text is input into the user interface. When sources are found on the database, a database application may be used to communicate between the database, the user interface  
25 and a data processor. The data processor processes and stores the linkage structure and any identified sources.

In step 204 of the method according to the present invention, the inputted text is published into a published document. The linkage structure is conserved and published with the published document as extra information to the inputted text, wherein the extra information  
30 corresponds to the recorded linkage metadata of the linkage structure. A reader of the published document may access the extra information by interacting with the published document. The extra information may include a summary of a selected portion of text, what parts of the text are inputted by the author, what parts are excerpts or quotes direct from sources, and information regarding the sources. Furthermore, the extra information may also  
35 include revision or tracked history data and suggestions for similar published documents and/or sources.

The reader is therefore provided with an information-rich publication with links to linkage metadata that is relevant to the topic of the published document. The reader is also provided with information regarding the author or authors' actions during the process of writing in step 203, and as such, can clearly audit the writing process of the author or authors.

5 The method according to the present invention thereof provides users with a more informative and transparent method of writing and publishing documents, with conserved linkage between author actions and information sources, which can be accessed again once the document has been published.

10 A more detailed method according to the invention is shown in Figure 10. In step 401 of the method according to Figure 10, the method includes presenting via a first user interface controls for a user to interact with a document database. In step 402 of Figure 10, the method includes presenting, via a second user interface, controls for a user to enter and record text based input into a document object or file format. In step 403 of Figure 10, the method includes capturing as metadata, the user's interactions with the document database and adding it to  
15 the document object or file format. In step 404 of Figure 10, the method includes capturing input text entered by the user and adding it to the document object or file format. In step 405 of Figure 10, the method includes determining an association between the text entered by the user and the captured metadata, and updating the metadata to reflect the association. In step  
20 406 of Figure 10, the method includes capturing an editing action from the user modifying text already entered by the user. In step 407 of Figure 10, the method includes adding metadata to the document object or file format to identify the modifications made to the text by the editing action. In step 408 of Figure 10, the method includes saving in memory the document object or file format, including the text entered by the user, the modifications made to the text by the editing action, and the metadata capturing the user interactions and the determined  
25 association.

The description above is intended to provide an illustration of an exemplary embodiment of the claimed invention, and not to limit the scope of protection afforded by the attached claims. Various modifications of the embodiment within the wording of the claims will occur to the skilled person.

30

35



## CLAIMS

1. A computer implemented method of capturing and preserving metadata in a document object or file format, the method comprising:
  - presenting via a first user interface controls for a user to interact with a document database;
  - presenting, via a second user interface, controls for a user to enter and record text based input into a document object or file format;
  - capturing as metadata the user's interactions with the document database and adding it to the document object or file format;
  - capturing input text entered by the user and adding it to the document object or file format;
  - determining an association between the text entered by the user and the captured metadata, and updating the metadata to reflect the association;
  - capturing an editing action from the user modifying text already entered by the user;
  - adding metadata to the document object or file format to identify the modifications made to the text by the editing action; and
  - saving in memory the document object or file format, including the text entered by the user, the modifications made to the text by the editing action, and the metadata capturing the user interactions and the determined association.
  
2. The computer implemented method of claim 1, wherein capturing text entered by the user and capturing as metadata the user's interactions with the document database are carried out simultaneously.
  
3. The computer implemented method of claim 1 or 2, wherein the interactions of the user, captured as metadata, include at least one of: a mouse click; a key press; a touch input on a touch screen; a scroll input; a website visit; a copy action; a paste action; a download; and a user-defined search of the database.
  
4. The computer implemented method of any preceding claim, wherein capturing input text comprises one or more of:
  - a first input method wherein text is identified as text freely typed by the user into the second user interface;
  - a second input method wherein text is identified as following a copy-and-paste action into the second user interface; and

a third input method wherein text is identified in a document file or a portion of a document uploaded by a user into the second user interface.

5. The computer implemented method of claim 4, wherein the determined association may identify one or more of text that is: a) manually input, b) copied from a document link, c) included in an uploaded document, d) edited text, e) unedited text, and f) entered during browsing.

6. The computer implemented method according to any preceding claim, wherein capturing as metadata further comprises:

identifying a source in the document database when the user interacts with the source in the document database; and

capturing, as metadata, source information from the source identified in the document database.

7. The computer implemented method of claim 6, wherein the source information includes at least one of:

an address or location of the source within the document database;

information content from the source;

a time and/or date when the source was accessed by the user.

8. The computer implemented method according to claim 6 or 7, wherein if the text based input is captured via the first input method, determining the association between the input text and the captured metadata comprises:

comparing the user-typed text with the captured metadata corresponding to the source information using textual based analysis of the text entered in the second user interface with text presented in the first user interface to determine identity or similarity between the texts.

9. The computer implemented method according to claim 6 or 7, wherein if the user input text is recorded via the second input method, determining the association between the input text and the captured metadata comprises:

capturing in the document object, as metadata, a copy action performed by the user interaction in the first user interface;

capturing in the document object, as metadata, a paste action of the copied text performed by the user interaction in the second user interface; and

updating the metadata to reflect the association.

10. The computer implemented method according to claim 6 or 7, wherein if the user input text is recorded via the third input method, determining the association between the input text and the captured metadata comprises:

capturing in the document object, as metadata, a document access action when the user accesses a document in the first user interface;

capturing in the document object, as metadata, a document or portion of a document upload action performed by the user interaction in the second user interface; and

updating the metadata to reflect the association.

11. The computer implemented method of any preceding claim, wherein adding metadata to the document object to identify the modifications made to the text by the editing action comprises either:

adding metadata indicating that text has been deleted, the deleted text being maintained in the document object or file format; or

removing metadata corresponding to the user input text when the input text is overwritten by the modified text.

12. The computer implemented method of any preceding claim, further comprising:

publishing the document object or file format in a published document;

wherein the published document includes a finalised text, the finalised text including:

when text is not modified by the user, the text entered by the user according to the text based input, and when text is modified by the user, the modified text;

wherein the published document also includes the metadata capturing the user interactions and the determined association.

13. The computer implemented method of any of claims 1 to 12, wherein the input text entered by the user includes numerical data; the numerical data being structured or unstructured data.

14. A computer system for capturing and preserving metadata in a document object or file format comprising:

a first user interface module including controls for a user to interact with a document database;

a second user interface module including controls for a user to enter and record text based input into the document object;

a data processing module; and

**a memory module;**

**wherein the data processing module is configured to access the document database to capture in the document object, as metadata, the interactions of the user with the document database;**

**wherein the second user interface module communicates with the data processing module to:**

**capture input text entered by the user in the user interface module and add it to the document object;**

**capture an editing action from the user modifying text already entered by the user and add metadata to the document object to identify the modifications made to the text by the editing action;**

**wherein the data processing module is configured to determine an association between the text entered by the user and the captured metadata, and to update the metadata to reflect the association;**

**wherein the memory module is configured to save the document object, including the text entered by the user, the modifications made to the text by the editing action, and the metadata capturing the user interactions and the determined association.**

**15. The system of claim 14, wherein the document database is either:**

**included in the system, wherein the first user interface module and the second user interface module are the same user interface module; or**

**separate from the system such that the system accesses the document database via a database application.**

**16. The system of claim 15, wherein, when the document database is separate from the system, the document database is accessed through a web-browser and the database application is a plugin or web-browser application.**

**17. The system of any of claims 14 to 16, further comprising a publishing module, wherein the publishing module is configured to:**

**publish the document object or file format in a published document;**

**wherein the published document includes a finalised text, the finalised text including:**

**when text is not modified by the user, the text entered by the user according to the text based input, and when text is modified by the user, the modified text;**

**wherein the published document also includes the metadata capturing the user interactions and the determined association.**

18. The system of any of claims 14 to 17, configured to perform the method of any of claims 1 to 13.

19. A non-transitory computer readable storage medium including instructions stored thereon executable by a processor, the instructions configured to carry out the method according to any of claims 1 to 13.

20. A computer implemented method of storing a file format for receiving and storing input data entered by a user of a computer terminal and metadata corresponding to characteristics of the input data, the file format configured to:

store an initial portion of input data entered by the user in a first portion of memory;

store, as metadata, characteristics of the initial portion of input data in a second portion of memory;

associate the initial portion of input data and the characteristics of the initial portion of input text; and

if a modification to the initial portion of input data is made forming a modified portion of input data:

store the modified portion of input data in the first portion of memory;

store, as metadata, characteristics of the modified portion of input data in a second portion of memory;

associate the modified portion of input data, the initial portion of input data, and the characteristics of the modified portion of input data;

wherein the characteristics of the input data comprise at least one of:

designation of the user who inputted the user input data;

a date and/or time when the user input data was inputted;

a source location indicating where the user obtained the user input data; and

an indication of the method of input the user performed to input the user input data.



**Application No:** GB1901830.8

**Examiner:** Mr Stephen Martin

**Claims searched:** 1-20

**Date of search:** 17 July 2020

**Patents Act 1977: Search Report under Section 17**

**Documents considered to be relevant:**

Category	Relevant to claims	Identity of document and passage or figure of particular relevance
X	1-20	WO2010/014403 A1 (CHI et al.) See in particular paragraphs 0003, 0007, 0009, 0035, 0038, 0046, 0047, 0048, 0056, 0058
X	1-20	US2008/0098317 A1 (CHEN et al.) See in particular paragraphs 0013, 0041, 0087, 0097
A		US2010/0030763 A1 (CHI et al.) see in particular paragraphs 0008, 0041, 0061, 0062
A		US2006/0218034 A1 (KELLY) See in particular paragraphs 0013, 0014, 0027, 0029, 0043, 0064, 0065

**Categories:**

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

**Field of Search:**

Search of GB, EP, WO & US patent documents classified in the following areas of the UKC<sup>X</sup> :

--

Worldwide search of patent documents classified in the following areas of the IPC

G06F
------

The following online and other databases have been used in the preparation of this search report

WPI, EPODOC, Patent Fulltext
------------------------------

**International Classification:**

Subclass	Subgroup	Valid From
G06F	0016/16	01/01/2019
G06F	0016/93	01/01/2019