



(43) International Publication Date
25 January 2007 (25.01.2007)

(10) International Publication Number
WO 2007/010187 A1

- (51) **International Patent Classification:**
G06F 17/30 (2006.01)

(21) **International Application Number:**
PCT/GB2006/002477

(22) **International Filing Date:** 4 July 2006 (04.07.2006)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
05254577.9 22 July 2005 (22.07.2005) EP

(71) **Applicant (for all designated States except US):** **BRITISH TELECOMMUNICATIONS PUBLIC LIMITED COMPANY** [GB/GB]; 81 Newgate Street, London, Greater London EC1A 7AJ (GB).

(72) **Inventors; and**

(75) **Inventors/Applicants (for US only):** **JOHNSTON, David, Alexander** [GB/GB]; 25 Haughgate Close, Woodbridge, Suffolk IP12 1LQ (GB). **MENDIS, Venura, Chakri** [LK/GB]; 56 Cemetery Road, Ipswich, Suffolk IP4 2HZ (GB). **SHEPPARD, Iain, Daniel, Mark** [GB/GB]; 7 Coppice End, Highwoods, Colchester, Essex CO4 9RQ (GB).

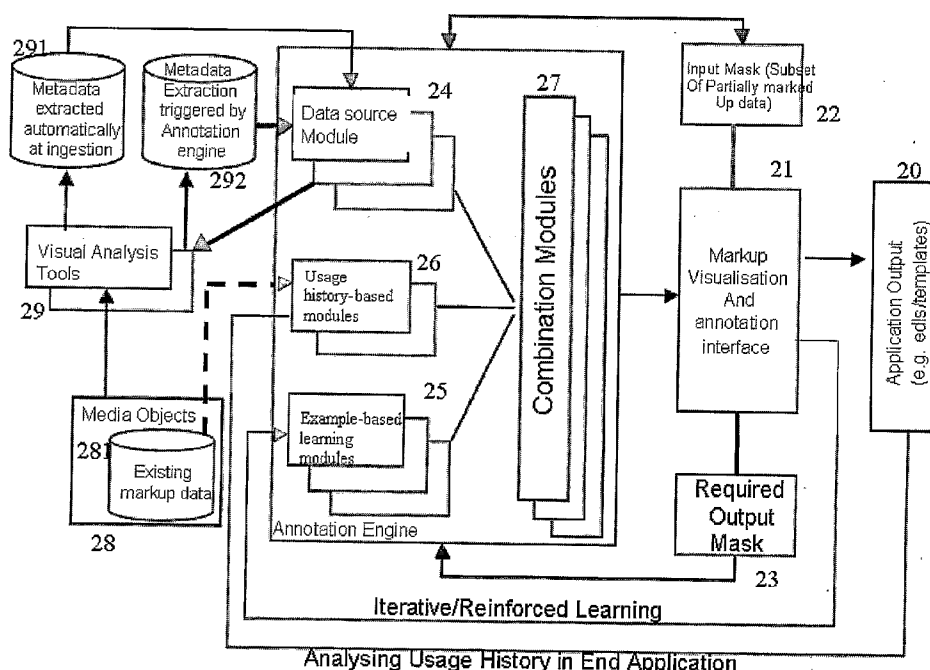
(74) **Agent:** **LIDBETTER, Timothy, Guy, Edwin;** BT GROUP LEGAL, INTELLECTUAL PROPERTY DEPARTMENT, PP: C5A, Bt Centre, 81 Newgate Street, London, Greater London EC1A 7AJ (GB).

(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, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, 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, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GO, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

- (54) Title:** DATA HANDLING SYSTEM



- (S7) Abstract:** Media objects (28) are organised for storage and subsequent retrieval by applying metadata tags in response to both a manual input (21) and at least one automated process (24, 25, 26) for analysing the media objects and the metadata tags applied thereto, the outputs of the processes being combined (27) to provide an output to the manual process. The process may be repeated iteratively



Published:

— with international search report

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.

Data Handling System

This invention relates to a data handling system, and in particular a device for organising and storing data for subsequent retrieval.

5 The advent of low cost digital cameras, cheap storage space and the vast quantity of media available has transformed the personal computer into a multi-purpose home entertainment centre. The versatile communications medium known as the "Internet" allows the ready transmission of recorded digital media files representing, for example, text, sound, pictures, moving images, numerical data, or software.

10 Note that although the plural 'media' is used throughout this specification, the term 'media file' is to be understood to include files which are intended to be conveyed to a user by only one medium – e.g. text or speech – as well as 'multimedia' files which convey information using a plurality of such media.

 The number of media files accessible via the Internet is very large, so it is
15 desirable to label them with a description of what they contain in order to allow efficient searching or cataloguing. Many users therefore add metadata to individual media files, or in media objects to be associated with the files. Metadata are data that relate to the content or context of the object and allow the data to be sorted – for example it is common for digital data of many kinds to have their date of creation, or their amendment history,
20 recorded in a form that can be retrieved. Thus, for example, HTML (HyperText Mark-up Language) files relating to an Internet web page may contain 'metadata' tags that include keywords indicating the subjects that are covered by the web-page. Alternatively, the keywords may be attached to a separate metadata object, which contains a reference to an address that allows access to the item of data itself. The metadata object may be
25 stored locally, or it may be accessible over a long-distance communications medium, such as the Internet. Such objects will be referred to herein as "media objects", to distinguish them from the actual media files to be found at the addresses indicated by the media objects. The expression 'media object' includes media data files, streams, or a set of pointers to a file or database. A media object contains a reference to a video/audio clip
30 and metadata about the clip.

 The structure of an individual media object consists of a number of metadata elements, which represent the various categories under which the object, (or more properly the information contained in the media file to which the object relates) may be classified. For example a series of video clips may have metadata elements relating to
35 "actors", "locations" "date of creation", and timing information such as "plot development"

or "playback order", etc. For each element, any given media object may be allocated one or more metadata values, or classification terms, from a vocabulary list of predefined values or enumerations for that metadata element; for example a list of "Actors" like "John", "Sarah", "Paul", "Rob". The vocabulary will, of course, vary from one element to another. In this specification, metadata conveying information such as actor, location, plot value, etc will be referred to as "formal" metadata, whilst time-related metadata, such as sequence, group, and cause/effect are referred to as "temporal" metadata.

The metadata elements and their vocabularies are selected by the user according to what terms he would find useful for the particular task in hand: for example the metadata values in the vocabulary for the metadata element relating to "actors" may be "Tom", "Dick", and "Harriet", those for "location" might include "interior of Tom's house", "Vienna street scene", and "beach", whilst those for "plot development" (temporal metadata) might include "Prologue", "Exposition", "Development", "Climax", "Denouement", and "Epilogue". Note that some metadata elements may take multiple values, for example two or more actors may appear in the same video clip. Others, such as location, may be mutually exclusive.

This approach has led to many useful applications in which stored metadata can be used to personalise content for the end user, automatically control network synchronisation, and many similar applications. A particular use of such metadata is described in International Patent specification WO2004/025508. This uses detailed formal and temporal metadata of the kind already described, (e.g. identifying individual actors or locations appearing in a video item, and time and sequence related information). A set of filters and combiners, hereinafter referred to as templates, are used to construct a narrative by arranging media objects in a desired sequence, by generating Edit Decision Lists (EDLs) - files containing a list of movie clips and the order in which to play them. These are essentially storylines generated by the media tools. They may also contain a separate audio track.

The use of metadata facilitates personalisation, cataloguing and searching. However, the application of many types of metadata is subjective and they can only be added by hand. This can be a tedious job, so in many cases users only apply the minimum amount of metadata to meet the application requirement. Unfortunately this approach leads to further problems when the database is subsequently to be used for a different purpose, or when databases are merged. Quite often they require marking-up all over again. To facilitate manual markup, a user interface that helps visualise existing levels of markup in a database and allow modification of the existing markup in the media

objects relative to other objects is described in the present applicant's International patent specification WO2005/086029, of which an embodiment of the user interface is shown in Figure 4 and will be described later.

However, applying metadata can be slow and tedious, especially with an already
5 large set of media objects and an application that requires detailed markup. Even when metadata already exists, it may be difficult to insert it easily and intelligently into new objects. There could also be several different sources or methods that can each be used to automatically obtain or extract the metadata, and a decision needs to be made on the relative validity of each data source. This problem can be further compounded when a
10 separately marked-up content database is imported, as reconciliation with existing markup may be difficult.

Processes exist that can be used to extract metadata from content, and automatically add this metadata to the media object. Such processes include artificial neural networks, and Data Clustering/ Data mining algorithms such as Minimal Spanning
15 tree algorithm and K- means algorithm. Expert systems are known that employ a number of such machine-learning algorithms and techniques to produce intelligent results or suggest markup. These data analysis techniques attempt to generalise specific manual markup cases so they can infer markup of other media objects. These AI data analysis techniques attempt to generalise specific manual markup cases so they can infer markup
20 of other media objects. Broadly speaking these traditional data analysis and machine learning systems fall into two categories. The first requires a large set of correct and/or incorrect data. The second attempts to model human behaviour over a long period of time by analysing manual markup.

Systems which use such automatic annotation, with or without a user interface to
25 allow the automatic process to be over-ridden and learnt from, are discussed in references such as US2005/114325 (Liu) and US2002/183984 (Deng)

Because media annotation in this context is extremely subjective it is difficult to provide a large enough example base for a machine learning system to produce suitable results for a number of different users of the system. Monitoring ongoing user activity is
30 also not sufficient, as markup pattern or habits will be specific to both the content used and the desired output, and so are likely to change from each project to the next.

One such system, described in the present applicant's International patent specification WO02/080027, uses a highlight detection module to fill the 'plot value' metadata element. Another system, described in GB2384078 (Fujitsu) describes a

visualisation and data-mining tool that analyses text and then generates captions for pieces of content.

These systems perform an offline analysis of the content and represent the information to the user. However, such processes are extremely complex and quite often
5 effective to only a limited specialised data set.

Another approach is to analyse how the partially marked up media objects are used, for example using a process in which media tools are used to generate storylines or sequences of video clips by querying formal metadata; the returned media objects are then arranged and sorted according to temporal metadata to form a storyline, by analysing
10 templates and the media objects pulled out by the templates.

As with many "Artificial Intelligence" systems it is often impossible or too complex to justify an automatic decision to a human user. Although any interface must attempt to communicate at least some of this information to the user, they do not let the user directly interact with the system. Moreover, the automated techniques described above are
15 specific to specialised data analysis applications, and have not been applied to more subjective and general media annotation processes. It is important to distinguish between using these techniques for the analysis of data and for the analysis of metadata. When analysing metadata, the way in which the metadata is used is an important additional component.

20 The present invention provides a data handling device for organising and storing media objects for subsequent retrieval, the media objects having associated metadata tags, the device comprising a user interface for allowing a user to apply metadata tags to some of the media objects, means for analysing the media objects and the metadata tags applied thereto, and means for automatically applying further metadata tags to other
25 media objects in the database in response to the said analysis.

The invention also provides a method of organising and storing media objects for subsequent retrieval, the media objects having associated metadata tags, wherein metadata tags are applied to some of the media objects in response to a manual input, comprising at least one automated process for analysing the media objects and the
30 metadata tags applied thereto, and for automatically applying further metadata tags to other media objects in the database in response to the said analysis.

The process may therefore be iterative, the automated process responding to corrections and amendments made manually, allowing the user to check and modify the output of the automatic markup, view results, and update or correct the results and then
35 allow the automated process to continue. There may be a plurality of automated

processes, selectable by the user, for applying metadata to the media objects, and their outputs may be selectively combined according to predetermined criteria. Metadata may be generated automatically by analysing content of the media objects themselves, for example by statistical analysis. There may also be a machine learning process for
5 analysing existing metadata and how the metadata is used in the end application.

The invention may be implemented by means of a computer program or suite of computer programs for use with one or more computers to perform or provide the invention.

This novel combination of existing data analysis and data mining techniques
10 allows media annotation to be supplied without interrupting the user's natural media annotation process. This invention allows the time spent performing markup to be minimised, making the process less tedious, by providing automated methods to use the metadata of existing objects to intelligently mark up new objects. Sometimes this may be as simple as spotting that all objects share a common metadata value, although it may
15 require the careful location of subtle patterns or the introduction of data from an external source. Although in many cases the metadata entered is subjective and is dependent on the editor using the system and on the sort of content the editor is trying to produce, automatically generated metadata for some fields can provide the user with a useful hint or guidance as to how other metadata elements should be filled. Hence a semi-automated
20 process that is a combination of automatic markup and manual markup is desired.

The preferred embodiment, to be described later, uses several automatic metadata generation techniques, and attempts to reconcile them based on a precedence order or statistical analysis. In addition, through an iterative process of manual markup, automatic markup, and correction of automatic markup, it attempts to simplify and hasten
25 the markup process. The markup process is iterative, with the user performing some manual markup, correcting some automatic markup, and then the system doing some automatic markup again. This process is repeated until the desired level of markup is achieved.

The invention may use external data sources produced by metadata extraction
30 using content analysis tools that generate metadata automatically. It may also use machine learning algorithms to analyse existing metadata and how this metadata is used in the end application. A user interface is provided so that the user can interact and perform manual markup, and recheck automatic markup and also drive the automatic markup by selecting input and output masks.

An embodiment of the invention will now be further described, by way of example only, with reference to the drawings, in which:

Figure 1 is a schematic diagram of a typical architecture for a computer on which software implementing the invention can be run;

5 Figure 2 is a schematic representation of the high level functional components that co-operate to perform this invention,

Figure 3 is a flow diagram showing how the invention may be performed

Figure 4 is a representation of a screen shot of the user interface.

Figure 1 shows the general arrangement of a computer suitable for running
10 software implementing the invention. The computer comprises a central processing unit (CPU) 10 for executing computer programs, and managing and controlling the operation of the computer. The CPU 10 is connected to a number of devices via a bus 11. These devices include a first storage device 12, for example a hard disk drive for storing system and application software, a second storage device 13 such as a floppy disk drive or
15 CD/DVD drive for reading data from and/or writing data to a removable storage medium and memory devices including ROM 14 and RAM 15. The computer further includes a network card 16 for interfacing to a network. The computer can also include user input/output devices such as a mouse 17 and keyboard 18 connected to the bus 11 via an input/output port 19, as well as a display 20. The person skilled in the art will understand
20 that the architecture described herein is not limiting, but is merely an example of a typical computer architecture. It will be further understood that the described computer has all the necessary operating system and application software to enable it to fulfil its purpose.

In the system described in the applicant's pending International patent application WO2004/025508 the editor manually adds some metadata either using a text-based
25 interface or a visual markup interface. He would then create templates, generate storylines and view the results. He can then iterate between the processes of markup, creating/modifying templates, and viewing the output storylines. This invention extends this traditional markup process by allowing automatic markup components to actively participate in the markup process. This adds some degree of complexity to the process
30 as these components require setting up, but this would only usually need to be done during the first iteration and would greatly accelerate the markup process on subsequent iterations. This process is illustrated in more detail in Figures 2 and 3.

Figure 2 is a schematic of the various system components which co-operate to form this embodiment. Media objects 28, which may already have some marking-up 281,
35 are analysed 29, 291, 292 and then marked up automatically 24 on the basis of that

analysis. The marked up objects are then assessed by a user through an interface 21, on the basis of "masks" 22, 23 indicative of the marking up to be done. This process then operates iteratively with a number of other automated annotation processes 25, 26, based on the interactions with the user interface 21 and the subsequent use 20 of the objects 28, the results of these processes being co-ordinated by a combination module 27. This embodiment provides three automated facilities 24, 25, 26 for handling a media object, but other facilities may be used instead. For example, suitable modules may be generated using the principles described in United States patent specification US2003/009469 (Platt), or that described by Naphade in "*Learning to annotate video databases*" published in the *Proceedings of SPIE* vol 4676, page 264.: (Conference on Storage and Retrieval for Media Databases: San Jose, CA: 23rd January 2002)

The present example uses a software object that contains a reference to a video/audio clip and contains metadata about it.

A Markup Visualisation and Annotation Interface 21 allows the user to view the media objects and their existing markup, and to modify them. This is illustrated in more detail in Figure 4, to be discussed later.

The operation of the system is illustrated in Figure 3.

The process starts 31 with a database 28 containing a number of media objects, collectively referenced 281. Adding a media object to a database involves creating a database object and adding a reference to the location of the actual media to generate a "thumbnail" image such as images 101-106 (Figure 4). The database may already include some marking up in the form of associated metadata.

The user first selects the metadata elements to be marked up (steps 32, 33), and creates an input mask 22 and an output mask 23. The masks 22, 23 control the annotation process, by specifying inputs and outputs to allow the user to filter his view of the data to identify the metadata currently of interest, as defined by the masks. When the user wishes to add markup to other metadata elements he can modify the input and output masks. The input mask relates to existing metadata: the output mask identifies the metadata elements the user desires to populate. The creation of the masks may be done manually, but it could be easily automated or partly-automated, in the case of the output mask by identifying metadata elements that are sparse with data.

Input Masks 22 contain references to metadata elements that already have data in them. These are generated either via manual markup from previous markup iteration, or automatically using values extracted by an external data source module or from a partially marked up imported database. The user selects these by using a simple tree control and

checkbox. These values can be pre-populated depending on metadata already present in the database, with the option for the user to remove or add metadata elements to the selection.

Output masks 23 are used in order to select the metadata element that the user desires to be filled in automatically. Similarly to input masks, the user selects these by using a simple tree control and checkbox. These values can be pre-populated by identifying metadata elements that are sparsely populated with data, with the option for the user to remove or add metadata elements to the selection

Having selected the metadata masks, the user next selects which annotation processes he wishes to use (step 34) and selects any variable properties for these processes (step 35). In the present example three processes 24, 25, 26, are available, which will each be described in detail later. These may obtain their data both from analysis of the media objects 21 themselves, or from the results of previous iterations of the process, as will be described.

The user also selects the manner in which the results of these processes 24, 25, 26 are to be combined (step 36), for example whether a simple logical "AND" or "OR" function (selection respectively by all, or by any, of these processes), or by "voting logic", or where one process carries greater weight than the others.

The sequence of processes 24, 25, 26, 27 is then performed on the data (step 37), and the user can then examine the results. If the user decides markup is incomplete he can update or modify the metadata element values specified by the output mask (step 38), and run the process again, specifying modifications to the input/output masks 22, 23 (steps 32, 33). Alternatively the user may go directly to the generation of storylines (step 39), and then update the filters (return to step 38).

In the case of a merged or partially marked up database, templates used to generate storylines and Edit Decision Lists (EDLs) would already exist, and the user may start this iterative process by viewing the existing data (step 39) before generating any more.

Once the user has confirmed or approved metadata generated by these modules 24, 25, 26, 27, this data is added to the existing metadata 21 and the process can continue iteratively.

The markup process 21 may be a manual process, but the preferred embodiment makes use of the Markup Visualisation and Annotation Interface 21 described in the applicant's International patent specification WO2005/086029, of which an embodiment of the user interface is shown in Figure 4, which provides a data handling device for

organising and storing media objects for subsequent retrieval, the media objects having associated metadata tags, comprising a display for displaying representations of the media objects, data storage means for allocating metadata tags to the media objects, an input device comprising means to allow a representation of a selected media object to be moved into a region of the display representing a selected set of metadata tags, and means for causing the selected set of tags to be applied to the media object.

The aforementioned patent application also provides a method of organising and storing media objects for subsequent retrieval, the media objects being represented in a display, and in which metadata tags are applied to the media objects by selecting an individual media object from the display, and causing a set of metadata tags to be applied to the selected media object by placing a representation of the selected media object in a region of the display selected to represent the set of tags to be applied.

As shown in Figure 4, the main difference of the process 21 from a conventional text-based or graphical markup interface is that it represents metadata in a media object in the context of metadata present in all the other media objects. In addition it allows easy modification or addition of metadata by simple drag and drop of icons representing media objects.

The process can be started in either of two modes – either directly acting on the entire database, or from a template view. The latter approach allows mark-up of a single cluster in the database, and segmentation of only the objects in that cluster. It also provides a less cluttered view for the user. In doing this it prevents unnecessary metadata from being added. Viewing this collection of objects within the visual marking-up tools lets the user easily visualise any metadata that might differentiate them, and if there is insufficient differentiation it allows the user to modify existing metadata to represent the object more accurately.

A metadata element, or combination of such elements, is selected by selecting one or more categories in a hierarchical structure. A control element of the user interface, containing a textual or graphical representation of the selections available, may be populated by appending an additional query to the template view described in the previous paragraph, identifying the media objects to which marking-up is to be applied. By populating the control elements in this manner it is possible to visualise the metadata marking-up in the entire database, or a defined subset of it, using any of the views described below. The media objects may be represented as either miniature images, as illustrated in Figure 4 ("Thumbnail" view) or as a plain text filename (Text-only view).

To add annotation to a particular element in the media object data model, it is first necessary to select the desired element in the hierarchical menu structure. As only a single class has been selected in this example, the extra steps and a set of List controls or boxes 400 – 411 are generated for each value in the vocabulary of available terms stored for each respective element. Existing metadata can be visualised by the manner in which media objects are arranged in the different boxes. Media objects that are in the 'unclassified' box 400 do not contain any metadata values for the particular element selected in the hierarchical menu structure. If the metadata values are not mutually exclusive, media objects may appear in more than one box.

10 The process requires a number of additional steps if more than one metadata tag has been selected. However, if only a single metadata tag has been selected, a single-metadata tag view is generated. In the example shown in Figure 4 the "Classification" element 41 has been selected under the "Creation" heading 40, and a number of metadata classes - "advertising" 401, "affairs" 402, "documentary" 403, "drama" 404, 15 "Education" 405, "Film" 406, etc are displayed as windows, each containing those media objects to which that metadata tag has been applied. Each individual media object 101, 102, etc is represented in the appropriate window by "thumbnail" images or other suitable means.

 The size of the display area representing each metadata tag is determined 20 according to the number of media objects to be displayed. Large groups, such as those shown at 401, 405, may be partially represented, with means 499 for scrolling through them. Thus a view of the various media objects 101, 102, 103, 104, 105, 106 etc, is generated, sorted according to the various "classification" categories (metadata values) 401, 402, 403 etc including an "unclassified" value 400. The view of existing metadata is 25 similar to a Venn diagram with non-overlapping sets. (If the sets were allowed to overlap – that is to say, one media object can have two or more metadata values applied to it – identical copies of the same object may appear in several of the boxes). All items are originally located in the "unclassified" set 400. The process of adding metadata to media objects relies on 'dragging and dropping' objects in and out of classification boxes, each 30 box being representative of a specific metadata value. The visual representation of the contents of each set makes it straightforward to ensure that similar media objects are placed in the same set (allocated the same tags). A user can also identify which categories are most heavily populated, and therefore worthy of subdivision. This would allow the distribution of the metadata in a database to be "flattened" i.e. all the media 35 objects should have similar amount of metadata associated with them.

The list controls are populated by inserting the metadata associated with the media object. To do this the user 'sorts' the media objects using a "drag and drop" function, for example using the left button of a computer "mouse". For the particular media object that is moved by that operation, the metadata value originally stored for it, which is
5 represented by the box in which it was originally located, is replaced by the value represented by the box to which it is moved. Moving from the "unclassified" area adds a value where none was previously recorded. If it is desired to delete a value, the object is moved from the box representing that value to the "unclassified" box.

In a single view, if the metadata element is extensible – that is to say, it can have
10 multiple values - such as "actors", moving to "unclassified" only removes one actor (the one it was moved from) - the others are unchanged. The icon that was moved to "unclassified" would be deleted if other values for the element still exist for that object. Deletion operates in a different way when multi-dimensional views are in use, as will be discussed later.

15 If a value is to be added, rather than replace an existing one, a "right click" drag-and-drop operation would generate a copy – in other words for that particular media object the metadata value represented by the box that it is "dropped" in is copied to that particular element but not deleted from the original element.

In this case a check is made to ensure that the "copy" operation is valid: in other
20 words to check that the origin and destination metadata elements are not mutually exclusive. If a multi dimensional view is being used, a check is also made as to whether the proposed move is between boxes that represent different metadata elements, and not just different values in the vocabulary of one metadata element. An error message is generated if such an attempt is made. Attempting to copy to or from the "unclassified"
25 area would also generate an error message.

The processor 10 automatically populates the list boxes 401, 402, 403, etc by querying the database 15 for media objects that contain the metadata values that are represented by the boxes to be displayed. The unclassified box is populated by running a NOT query for all the metadata values.

30 In the single-dimensional view shown in Figure 4, the user selects a single metadata element 41 from the hierarchical menu structure. In the illustrated example this is the "classification" element 41. The media objects are now sorted according to the vocabulary values of their respective metadata elements – for example, "advertising" 401, "affairs" 402, "documentary" 403, "Drama" 404, "Education" 405, "Film" 406 etc. An empty
35 box may appear – this denotes a metadata value defined in the database vocabulary but

not used by any of the objects. An unclassified box 400 contains media objects (e.g. 100) that do not contain any values for the selected metadata element.

More than one metadata element may be selected by identifying more than one checkbox. The user may select whether to display a multi dimensional view or an
5 "Intersection" view.

By identifying the boxes that contain the most objects, and which ones have the least or even none, the user can decide which vocabulary terms are most relevant to a database and get a much better idea of how much more marking-up he needs to do. Moreover, multi-dimensional and intersection views can be used to afford visual hints to
10 the user to fill in missing metadata values. The user can graphically see which metadata element or values produce which cluster. By using this in combination with a template the user can easily generate the movie clips he desires by adding markup just to the clips he is interested in. The user can see the markup of an object relative to all the others in the database, and so he is given an idea of how much more markup is needed to add it into a
15 cluster or differentiate it when it is in a cluster.

The embodiment provides three main facilities for handling a media object. In this example it is a software object that contains a reference to a video/audio clip and contains metadata about it.

Media objects to which no value has yet been applied for this specified metadata
20 element are displayed in a separate field 400. A second facility allows unmarked media objects to be identified so that the user can perform the marking-up operation.

A third facility allows a set of media objects that have one or more common elements to be identified, and allows the user to separate or differentiate the media objects by adding new or different metadata, or to find some other criteria that achieve a
25 further distinction between the media objects.

Note that some of the regions may represent sets have only one member, or none at all (the "empty" set). Other regions may represent Boolean operators such as "intersection" sets (sets of objects each having two or more specified metadata tags) or "union" sets (sets of objects having one of more of a specified group of such metadata
30 tags).

This process 21 guides the user to complete the minimum necessary marking-up to accomplish his task, and to provide a more even distribution of metadata in the media objects, by providing visual cues to help complete missing marking-up information and also provide a visual representation of the existing level of marking-up or completeness of
35 a database. In particular, it may provide an indication of media objects for which no

marking-up information has yet been applied for a particular element. In the described embodiment this takes the form of a display area in which "unclassified" objects are to be found.

In a large database, there are likely to be too many individual elements in the metadata structure, or values in their vocabularies, to allow all of them to be displayed simultaneously. In the preferred arrangement an hierarchical structure is used to allow the user to select objects having specified elements and values, thus making the interface less cluttered by letting the user view only those metadata objects having elements in which he is currently interested. He may then sort them using different search terms (elements).

In the visual markup interface, the media object location 401, 402, etc on the user interface 400 is used to indicate what metadata is present. For example, if a media object is in list control titled "actor->john", the metadata value for metadata element "actor" is "john". If it is in "actor->unclassified" it has no value in the metadata element "actor". In the present invention the visual markup interface is modified to provide the user with an indication of automatic markups that have been applied. Essentially it attempts to provide the user with a justification of why the system has modified or added a metadata value to a media object. This is done by highlighting such media objects so they can be differentiated from the manually marked up objects. If the user clicks on one of these objects an arrow appears indicating where the object was (in other words, what metadata, if any, had already been applied to it) prior to automatic markup. A double click can bring up more detailed information, indicating which module made the decision and, if several were used, a brief explanation of the combination module used to make final decision.

These automatic metadata values can be accepted in bulk by clicking on an 'accept all' button. It is possible to revert to the automatically modified data back to its previous state, for example by right-clicking on a highlighted media object.

Output and input masks do not need to be mutually exclusive. If a conflict occurs (i.e. automatic markup attempts to modify an existing metadata value) the media object, metadata element and values in question is flagged to the user by the Annotation Interface 21.

This embodiment uses a plug-in architecture. As seen in Figure 2, the modules 24, 25, 26 are packaged software plug-ins, capable of inferencing and/or extracting data from an external source and updating metadata values for a given mask. This allows several different plug-in modules to be used together and testing to be carried out to identify the most suitable for a given application. It also allows the easy integration of

further applications and algorithms. Several plug-ins can be used together and results from several of these can be reconciled to produce more accurate results, as will be described in more detail with reference to the combination module 27. In this embodiment three varieties of module are used in this implementation, and they will each now be
5 described.

The first is a data source module 24 having a highlight detection algorithm to automatically extract plot values for all media objects in the database. Such a process is known from International patent Specification WO02/080027. This runs as a separate offline process and generates a text file with the relevant metadata. One of the properties
10 that need to be set up in the data source module is the location of this text file. This highlight detection process may be run offline at the content ingestion stage; however such modules could also be triggered as part of an automatic markup process initiated by the data source module if they can run faster than real time.

As shown in Figure 2, existing annotation data 21, which has already been added
15 as part of the current annotation session or the result of merging with an old database, is used by the example-based learning module 25. The visual analysis tool 29 analyses the content of the media objects themselves, for example using a highlight detection algorithm to suggest plot values. The visual analysis tool 29 can be implemented either as an offline process that generates metadata at content ingestion, or as a realtime process
20 triggered by a data source module 24.

The example based learning module 25 uses input and output mask values and also a collection of media objects to use as correct examples for training the system to perform further mark ups. One possible implementation is described in an article by D. Nauck, M Spott and B Azvine Novel Data Analysis tool published in the British
25 Telecommunications Technology Journal Volume 21, No 4, October 2003. This disguises complex settings by automatically adjusting the setting for a given data set according to a simple user-supplied criteria.

The Usage Analysis module 26 is similar to the Example based learning module 25 in that they both use machine learning. They differ in that example based learning
30 modules 25 use only exiting metadata as inputs, while the usage analysis module 26 uses usage data, such as EDLs and resulting storylines generated as an additional input. Moreover, they do not require a list of media as training examples. This category of modules would be most useful when merging databases that already have existing templates.

The Combination Module 27 combines the results generated by the modules 24, 25, 26 discussed above. It may be set up to resolve conflicts between the outputs of the different modules by establishing a precedence order, for example giving the data source module 24 priority over the others. Alternatively, a statistical analysis may be performed
5 based on the accuracy for each of the outputs of all the modules could be carried out, based on how much modification is done to the outputs of each module by the user in the human interface 21. A preferred method uses the mode of the results of all the modules 24, 25, 26 as the final output decision – in other words no module has precedence, but in the event of a conflict, similar results from any two modules will prevail over a conflicting
10 output from a single module.

The output metadata are displayed to the user using the visual markup interface 21, and the user has the opportunity to override the automatically extracted data.

The invention may be used to compile a media article such as a television programme using a wide variety of media, such as text, voice, other sound, graphical
15 information, still pictures and moving images. In many applications it would be desirable to personalise the media experience for the user, to generate a “bespoke” media article.

Digital sets of stored media objects can be stored in a variety of formats, for example, a file may merely contain data on the position of components seen by a user when playing a computer game – the data in that file subsequently being processed by
20 rendering software to generate an image for display to the user.

As will be understood by those skilled in the art, the invention may be implemented in software, any or all of which may be contained on various transmission and/or storage mediums such as a floppy disc, CD-ROM, or magnetic tape so that the program can be loaded onto one or more general purpose computers or could be
25 downloaded over a computer network using a suitable transmission medium. The computer program product used to implement the invention may be embodied on any suitable carrier readable by a suitable computer input device, such as CD-ROM, optically readable marks, magnetic media, punched card or tape, or on an electromagnetic or optical signal.

CLAIMS

1. A data handling device for organising and storing media objects for subsequent retrieval, the media objects having associated metadata tags, the device comprising a user interface for allowing a user to apply metadata tags to some of the media objects, means for analysing the media objects and the metadata tags applied thereto, and means for automatically applying further metadata tags to other media objects in the database in response to the said analysis
2. A data handling device in accordance with claim 1, further comprising a plurality of tag application means for the automated application of metadata tags to media objects.
3. A data handling device according to claim 2, further comprising means for allowing a user to select the tag application means to be used.
4. A data handling device according to claim 2 or 3, further comprising means for selectively combining the outputs of the plurality of tag application means according to predetermined criteria.
5. A data handling device according to any preceding claim, comprising content analysis means for generating metadata automatically by analysing content.
6. A data handling device according to any preceding claim, comprising machine learning means for analysing existing metadata and how the metadata is used in the end application.
7. A data handling device according to any preceding claim, wherein the user interface has means to allow a user to check and modify the output of the the automatic markup means.
8. A computer program or suite of computer programs for use with one or more computers to provide any of the apparatus as set out in any one of claims 1 to 7.
9. A method of organising and storing media objects for subsequent retrieval, the media objects having associated metadata tags, wherein metadata tags are applied to

some of the media objects in response to a manual input, comprising at least one automated process for analysing the media objects and the metadata tags applied thereto, and for automatically applying further metadata tags to other media objects in the database in response to the said analysis

5

10. A method according to claim 9, wherein the process is iterative, the automated process responding to corrections and amendments made manually.

11. A method according to claim 9 or 10, wherein there are a plurality of automated
10 processes that may be selected to apply the metadata to media objects.

12. A method according to claim 11, wherein the outputs of the automated processes may be selectively combined according to predetermined criteria.

15 13. A method according to claim 9, 10, 11 or 12, wherein metadata are generated automatically by analysing content of the media objects themselves.

14. A method according to claim 9, 10, 11, 12 or 13, comprising a learning process for analysing existing metadata and how the metadata is used in the end application.

20

15. A computer program or suite of computer programs for use with one or more computers to provide the method of any one of claims 9 to 15.

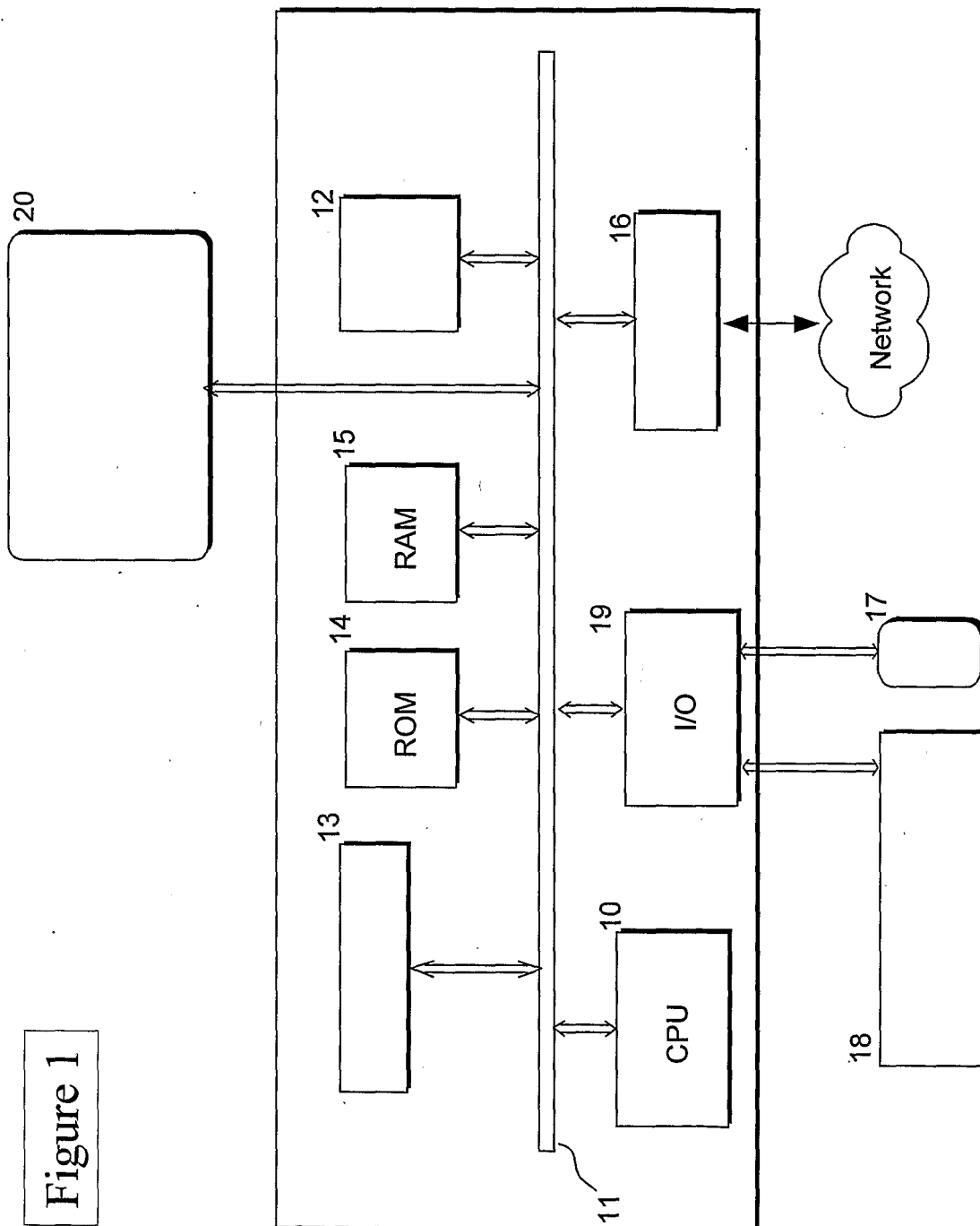
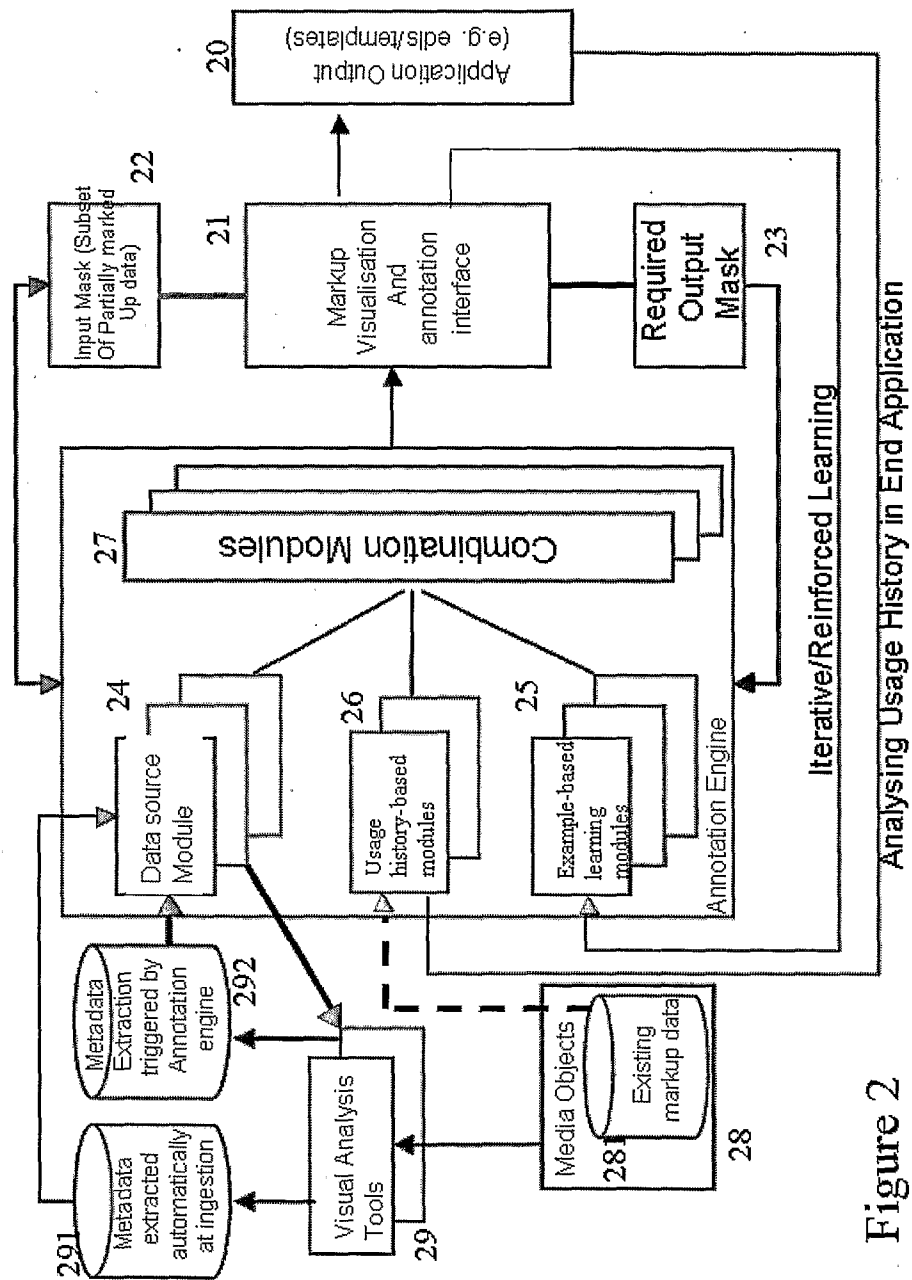


Figure 1



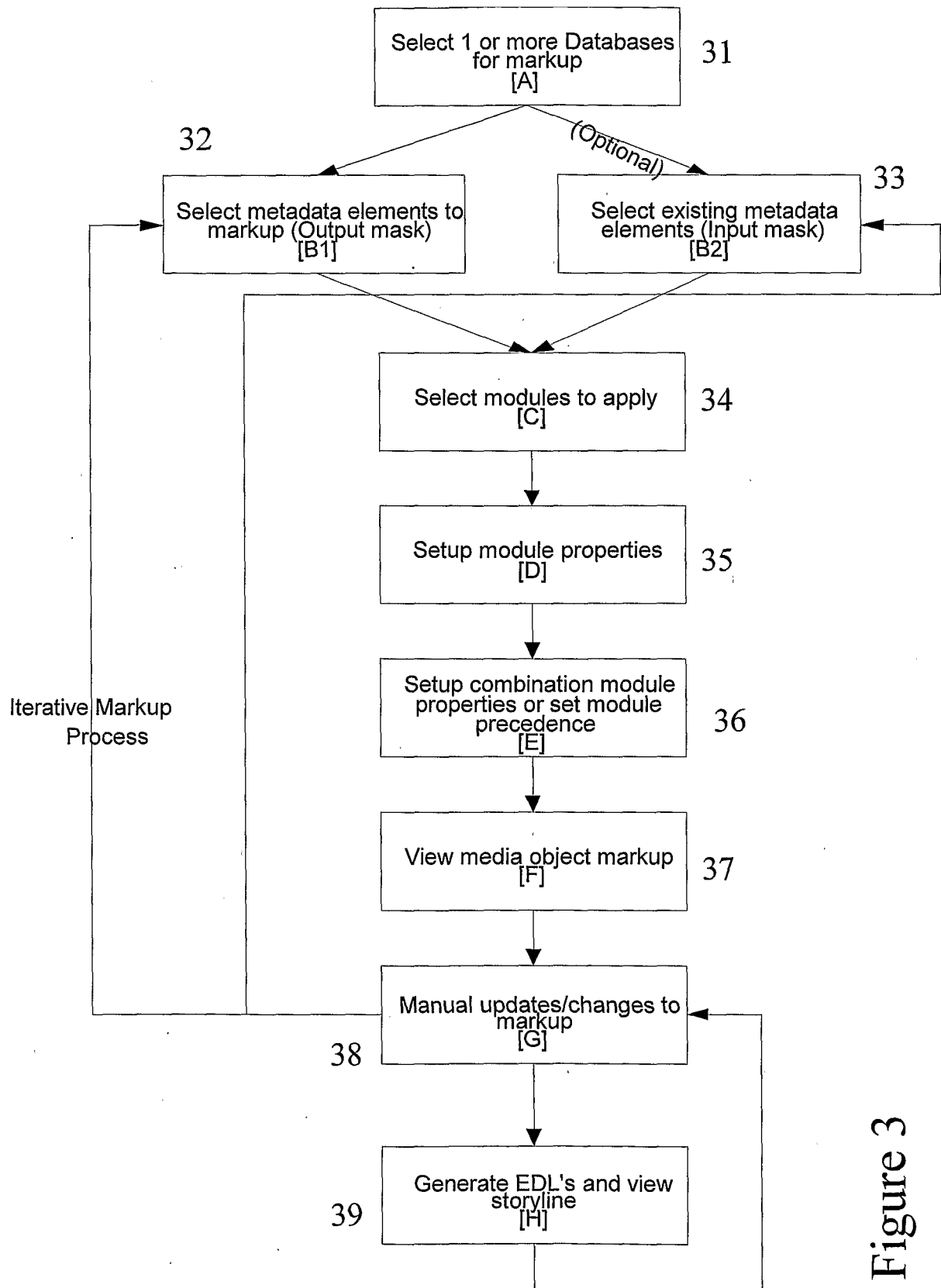


Figure 3

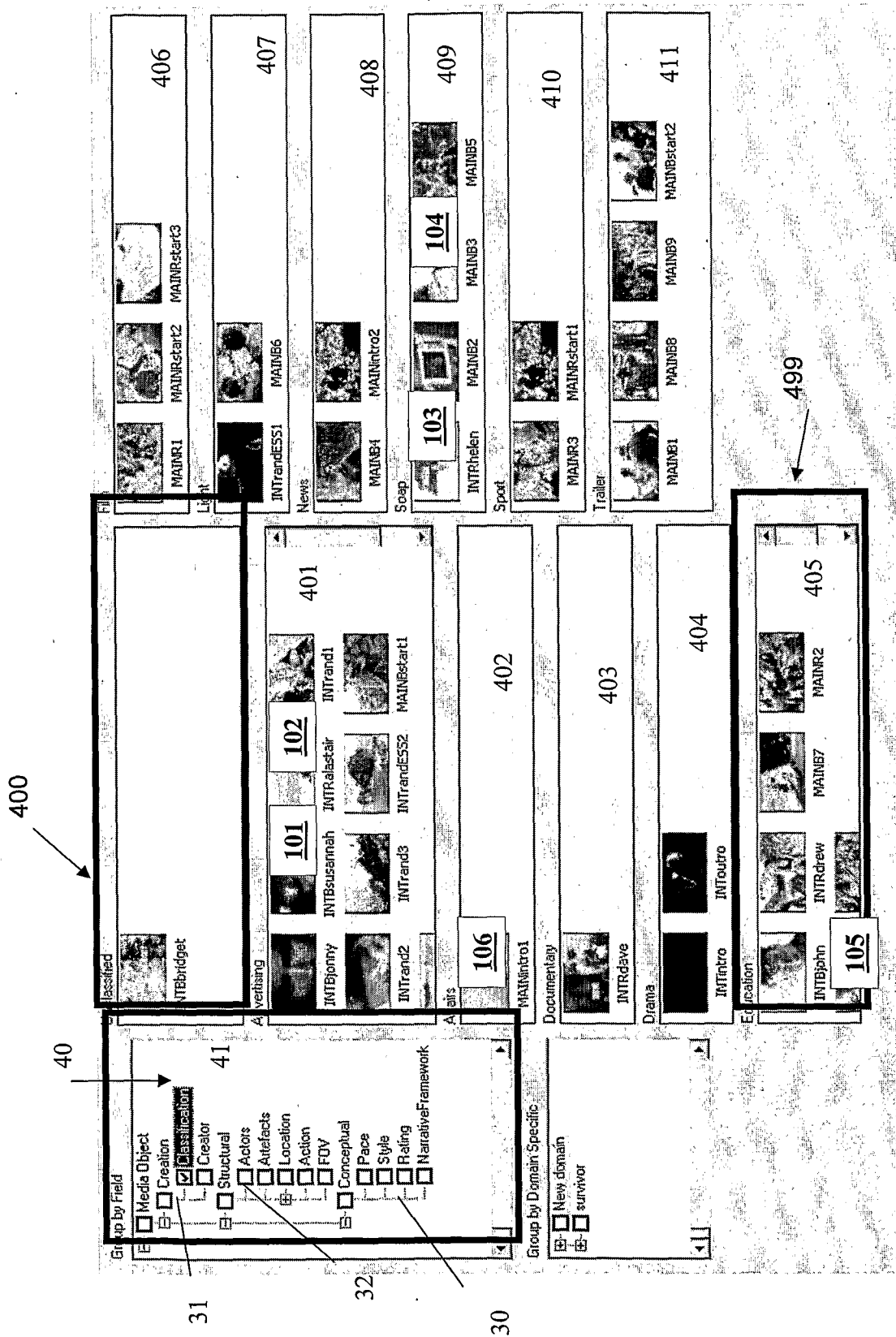


Figure 4

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2006/002477

A. CLASSIFICATION OF SUBJECT MATTER
INV. 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)
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, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2005/114325 A1 (LIU WEN-YIN ET AL) 26 May 2005 (2005-05-26) cited in the application abstract; figures 2-7 paragraphs [0009] - [0014] paragraphs [0037] - [0041] paragraphs [0051], [0059], [0060] paragraphs [0071] - [0078] paragraphs [0111] - [0116] ----- -/--	1-15

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

☒ See patent family 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

29 August 2006

Date of mailing of the international search report

07/09/2006

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

Laurentowski, A

INTERNATIONAL SEARCH REPORT

International application No

PCT/GB2006/002477

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CHING-YUNG LIN ET AL: "VideoAL: a novel end-to-end MPEG-7 video automatic labeling system" PROCEEDINGS 2003 INTERNATIONAL CONFERENCE ON IMAGE PROCESSING. ICIP-2003. BARCELONA, SPAIN, SEPT. 14 - 17, 2003, IEEE, USA, vol. 2, 14 September 2003 (2003-09-14), pages 53-56, XP010670367 ISBN: 0-7803-7750-8 abstract; figures 1,3,4	1-15
X	SARVAS R ET AL.: "Metadata creation system for mobile images" PROCEEDINGS OF THE 2ND INTERNATIONAL CONFERENCE ON MOBILE SYSTEMS, APPLICATIONS AND SERVICES, MOBISYS'04, BOSTON, MA, USA, 6 June 2004 (2004-06-06), pages 36-48, XP002393963 ACM Press, USA ISBN: 1-58113-793-1 abstract; figures 2-4 page 36, paragraph 1 - page 41, paragraph 2.5 pages 42,43, paragraphs 3.1,3.2 page 44, paragraphs 3.4.4,3.4.5 pages 45-46, paragraphs 4.3,5	1-15
X	DAVIS M ET AL: "From Context to Content: Leveraging Context to Infer Media Metadata" PROCEEDINGS OF THE 12TH ANNUAL ACM INTERNATIONAL CONFERENCE ON MULTIMEDIA, NEW YORK, USA, 10 October 2004 (2004-10-10), pages 1-8, XP002374239 ACM Press, USA ISBN: 1-58113-893-8 abstract; figures 1,2 pages 1-2, paragraph 1 page 3, paragraph 3 - page 6, paragraph 3.5 pages 6-8, paragraphs 4.3,5	1-15
X	WENYIN L ET AL: "Semi-automatic image annotation" PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON HUMAN-COMPUTER INTERACTION, AMSTERDAM, NL, 9 July 2001 (2001-07-09), pages 326-333, XP002258647 cited in the application abstract; figure 1 pages 327-328, paragraph 2.1 pages 328-329, paragraph 2.3-2.4	1-15

-/--

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2006/002477

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2002/183984 A1 (DENG YINING ET AL) 5 December 2002 (2002-12-05) cited in the application abstract; figures 1-13 paragraphs [0007] - [0062] -----	1-15
X	NAPHADE M. ET AL: "Learning to annotate video databases" PROCEEDINGS OF SPIE, vol. 4676, 23 January 2002 (2002-01-23), pages 264-275, XP002349160 Conference on Storage and Retrieval for Media Databases 2002, San Jose, CA, USA cited in the application abstract page 264, paragraph 1 - page 266, paragraph 2.1 page 272, paragraph 4 - page 274, paragraph 5 -----	1-15
X	US 2003/009469 A1 (PLATT JOHN CARLTON ET AL) 9 January 2003 (2003-01-09) cited in the application abstract; figures 3-7,10,11 paragraphs [0035] - [0043] paragraphs [0046] - [0054] paragraphs [0064] - [0076] -----	1-5, 7-13,15

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/GB2006/002477

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2005114325 A1	26-05-2005	US 6970860 B1	29-11-2005
		US 2005010553 A1	13-01-2005
US 2002183984 A1	05-12-2002	AU 2002305841 A1	16-12-2002
		EP 1419458 A2	19-05-2004
		JP 2005518001 T	16-06-2005
		TW 223171 B	01-11-2004
		WO 02099703 A2	12-12-2002
US 2003009469 A1	09-01-2003	US 2006074973 A1	06-04-2006