

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
15 November 2007 (15.11.2007)

PCT

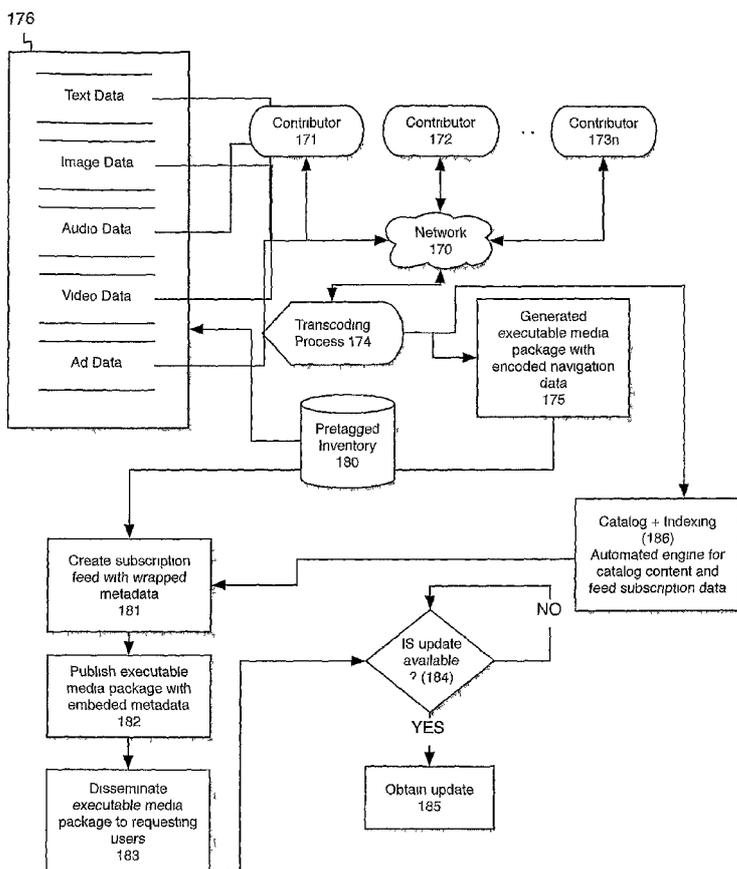
(10) International Publication Number
WO 2007/130116 A1

- (51) **International Patent Classification:**
G06F 11/00 (2006.01)
- (21) **International Application Number:**
PCT/US2006/035989
- (22) **International Filing Date:**
13 September 2006 (13.09.2006)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
60/797,336 2 May 2006 (02.05.2006) US
- (71) **Applicant (for all designated States except US):** DIGI-CORP, INC.; 4143 Glencoe Ave, Unit B, Marina Del Rey, CA 90292 (US).
- (72) **Inventors; and**
- (75) **Inventors/Applicants (for US only):** RIFKIN, Jay [US/US]; 1723 Cloverfield Blvd, Santa Monica, CA 90404 (US). GATCH, PhU [US/US]; P.O.Box 25046, Los Angeles, CA 90025 (US).

- (74) **Agent:** MAYO, Joseph, J.; DALINA LAW GROUP PC, 7910 Ivanhoe Ave #325, La Jolla, CA 92037 (US).
- (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, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, 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, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) **Title:** SYSTEM AND METHOD FOR ASSEMBLING DATA



(57) **Abstract:** One or more embodiments of the invention are directed to a system and method for assembling data from multiple network sources into executable media packages configured for targeted dissemination to an audience of requesting viewers. Gathering of advertisement viewing information and click-thrus is obtained by layering scripts into the executable media package to determine if advertisements have been viewed. A free product or service may be provided to a user that has clicked thru a link, with compensation from the content producer to the merchant for the product and compensation from the advertiser to the content producer which yields a net profit.

WO 2007/130116 A1



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.

SYSTEM AND METHOD FOR ASSEMBLING DATA**INVENTORS:**Jay **Rifkin****Phil Gatch****BACKGROUND OF THE INVENTIO****FIELD OF THE INVENTION**

[0001] Embodiments of the invention relate to the field of computer software and hardware system. More particularly but not by way of limitation, embodiments of the invention are directed to a system and method for assembling data from multiple network sources into executable media packages configured for targeted dissemination to an audience of requesting viewers which may include advertisement tracking and incentivized click-thrus.

DESCRIPTION OF THE RELATED ART

[0002] Modern communication systems provide an efficient mechanism for rapidly communicating data to a universe of multiple recipients. Giving users such capabilities has dramatically changed existing distribution systems for digital media. Radio and television for instance are becoming replaced by other technologies. In a prior era music was distributed in the form of Compact Discs (CDs) and made available to consumers via physical storefronts or simply played to customers on radio stations. The digital data stored on these CDs remained largely undistributed across computer networks because doing so was overly complex and largely impractical due to the large file size required to maintain a relatively decent level of audio quality. The advent of compressed file formats for audio data such as the MPEG Audio Layer 3 format (commonly referred to as MP3) resulted in smaller file sizes and thereby eased the constraints that previously prevented the distribution of music across computer networks. The smaller files sizes are a result of the way data is stored in the MP3 file (as pulse-code modulation-encoded (PCM) audio data where portions that are considered less important to

human hearing are discarded). Other audio data formats such as Advanced Audio Coding (AAC) which is also known as MPEG-2 Part 7 also provide a compressed format for distributing music files.

[0003] Once consumers had the ability to encode audio data into these and other compressed file formats music sharing became significantly more commonplace as users could now email or share music data across a computer network. Systems such as Napster, Kazza and Grokster which in essence provide users with a way to share music stored on one computer with users on another computer. Such systems are referred to as peer-to-peer systems (P2P). A pure P2P network is a network that relies on the computer power and bandwidth of each computer / peer that is connected to the network. A P2P network does not have the notion of clients or servers but has each peer function as both a client and a server. This model of network arrangement differs from the client-server model where communication is usually to and from a central server. Peer-to-peer architecture embodies one of the key technical concepts of the Internet, described in the first internet Request for Comments, "RFC 1, Host Software." More recently, the peer-to-peer concept achieved wide spread recognition in the general public in the context of the absence of central indexing servers in architectures used for exchanging multimedia files such as audio data.

[0004] The popularity of peer-to-peer systems, which offered the lure of free music but had legal issues about the legitimacy of such distribution, led to more commercially acceptable alternatives such as Apple Computer's iTunes™. Music distribution systems such as Apple Computer's iTunes™ system brought legitimacy to a form of distribution that was once viewed negatively by the entertainment industry. Using data distribution systems such as iTunes™ users can now purchase and download music and videos of their liking with a few simple mouse clicks. Another option users have is receive a subscription feed of automatically delivered audio or video content. These subscription feeds are accomplished using either Really Simply Syndication (RSS) or Atom syndication for listening on mobile devices and personal computers and are called podcasts usually featuring audio or video versions of a show, lesson or any other type of periodic broadcast. Subscribers to a podcast typically receive updates at sporadic or planned intervals such as daily, weekly, etc. There are also podcast networks that feature multiple shows on the same feed. Podcasting is about giving users customization and control over the audio and video data the user desires. In essence the goal in podcasting is to create audio

or video content for an audience that wants to listen to the content when they want, where they want and how they want.

[0005] As devices such as the iPod™ or other MP3 type players become increasingly popular distribution mechanisms such as podcasting must become more sophisticated. Current tools for authoring podcasts or other RSS feeds are cumbersome and lack the functionality needed to be useful for large-scale productions. Moreover the process of creating production quality podcasts is unnecessarily complex and existing software solutions are not tailored towards situations having more than one author for a given podcast. Thus prior art systems are deficient of an effective mechanism for generating podcasts based on collective input from a plurality of users. For instance, existing systems do not have effective mechanisms for assembling a podcast or other type of media package in cases where a plurality of contributing users provide input into what data the podcast is to contain. Another problem existing solutions have is that there is no effective way to manage the podcast content based on audience characteristics such as demographics or geographic location. For instance, the same content is typically distributed to all users who subscribe to the podcast without further customization based on preferences or likely preferences of the user. Known systems do not allow for podcast related advertisement tracking or podcast related free download of media to incentivize click-throughs. To overcome the problems stated above and others not explicitly set forth there is a need for an innovative system and method for assembling data from multiple network sources into executable media packages configured for targeted dissemination to an audience of requesting viewers.

BRIEF SUMMARY OF THE INVENTION

[0006] Embodiments of the invention provide users with an innovative way to create and distribute executable media packages. Systems designed to implement one or more attributes of the invention configured to create or modify an executable media package based upon input obtained from a plurality of contributing users. Hence the system provides an enterprise solution that can scale to any number of users. Distribution of the executable media package is achieved using any communication and computational mechanism that enables users to obtain and view the contents of the executable media package. In at least one case the executable media package comprises data such as podcast data or other data obtainable via a subscription feed such as Really Simple Syndication (RSS) protocol. The podcast data is assembled from media data and generated in one embodiment of the invention via transcoding process that outputs a generated podcast file with encoded chapter data and metadata.

[0007] The system is configured to accept media data that contains pretagged inventory where the pretagged inventory is associated with a specific user attribute such as a demographic or geographic characteristic. An advertisement targeted at a specific demographic group is an example of pretagged inventory. The subscription feed is generated as part of the publication process and generally summarizes the content of the media data. Hence the subscription feed is what provides users who wish to execute the executable media package with a way to subscribe to a specific set of executable media packages. Once an executable media package is assembled for subscription embodiments of the invention provide a mechanism for distributing the executable media package to achieve a business goal such as the desire to have viewers within a specific demographic access to the message contained therein. This distribution occurs when the executable media package is published along with the embedded metadata. Once published the executable media data is disseminated to requesting users (e.g., those subscribed to the feed) upon request.

[0008] One or more embodiments of the invention enable the tracking of advertisement viewing. Using layering for example in a podcast file allows hidden functionality to be bundled into the podcast. Interactive Flash, pictures, audio, advertisements and scripts may be embedded into a podcast. Once embedded, a script may be utilized to record the IP address and time and date of the tracked event. For example, a MOV file embedded into a low layer that is really a script,

generates an error which allows the script to record the IP address and time and date of the execution of the script. By timestamping events, the verification of advertisement downloading and viewing is achieved. The information related to the advertisement viewing may be tracked and reported to an advertiser or other content provider. In addition, specific targeting of advertisements may occur when a user opts into a profiling form, or when the user purchases something for example.

[0009] In addition to tracking podcast related advertisement viewing, one or more embodiments of the invention enable incentivized click-thru functionality. For example, by presenting an advertisement embedded into a layered podcast, obtaining a click-thru, redirecting to a merchant who delivers free media content, incentivized click-thru functionality is achieved. Using these embodiments of the system allows for the generation of income when content producers or advertisement providers pay a merchant for click-thrus. In certain countries where media is generally not paid for, any free giveaway of media that results in payment to the content creators provides an incentive for content creators to enter that market. For example, in China a click-thru cost of 63 cents with a cost per song of under 8 cents provides a large incentive for using an embodiment of the invention compared to the United States market where a click-thru cost of 1 dollar with a cost per song of 60 cents providing a lower, yet positive incentive. Since certain countries allow for exclusive rights, for example for podcast rights, a percentage of click-thru revenue may be contracted to a granting authority for the exclusive license to provide the system enabled herein.

[0010] Ad content may be auctioned out allowing for different media buyers to provide ads that dynamically change based on any variable, such as the time of viewing, after a number of ads has been displayed or based on the profile of the user viewing the ad or any other characteristic associated with the user, location of user, time, or any other variable.

DESCRIPTION OF THE DRAWINGS

[0011] The figures referred to throughout and setting forth one or more example embodiments of an innovative system and method for assembling data from multiple network sources into executable media packages configured for targeted dissemination to an audience of requesting viewers are as follows:

[0012] Figure IA is a block diagram showing a system and method for assembling data from multiple network sources into executable media packages in accordance with one or more embodiments of the invention.

[0013] Figure IB is a high-level block diagram that illustrates the basic functionalities required to implement a solution that assembles data from multiple networked sources into media content for purposes of distributing said media content to a targeted audience of requesting users.

[0014] Figure 1C provides an illustration of the methodology implemented by one or more embodiments of the invention via a chaptering / metadata definition interface.

[0015] Figure 2 provides an example of the media collection interface configured to provide upload functionality in accordance with one or more embodiments of the invention.

[0016] Figure 3 illustrates a chaptering interface that provides users with a tool for inserting navigation data and other metadata such as hyperlink data and images.

[0017] Figure 4 illustrates a media management interface for enabling a plurality of users to access and manage the media data pool.

[0018] Figure 5 illustrates a publishing interface for enabling authorized users to publish executable media packages to a subscription feed.

[0019] Figure 6 illustrates a search interface that utilizes indexed data associated with executable media packages.

[0020] Figure 7 illustrates channel or episode specific advertising in accordance with one or more embodiments of the invention.

[0021] Figure 8 illustrates an executable media package in the form of a podcast that is configured to save click-thru information and provide a user interface element that results in a free product giveaway if the user clicks-thru the link.

DETAILED DESCRIPTION OF THE INVENTION

[0022] An innovative system and method for assembling data from multiple network sources into executable media packages configured for targeted dissemination to an audience of requesting viewers will now be described. In one or more embodiments of the invention the content contained in the media packages is customizable based on identified attributes of the targeted viewer. In the following description numerous specific details are set forth in order to provide a more thorough understanding of embodiments of the invention. It will be apparent, however, to an artisan of ordinary skill that the description given is an example and that the present invention may be practiced without incorporating all aspects of the specific details described herein. For instance systems embodying one or more aspects of the invention can be implemented using a number of different technologies and hence the invention is not to be limited to the specific examples given herein but can also include other implementations used to accomplish the functions described herein. In some cases specific features well known to those of ordinary skill in the art have not been described in detail so as not to obscure the invention. Readers should note that although examples of the invention are set forth herein, the claims, and the full scope of any equivalents, are what define the invention.

[0023] Conceptual Overview

[0024] Embodiments of the invention provide users with an innovative way to create and distribute executable media packages. For instance, systems designed to implement one or more attributes of the invention are coupled to an interconnection fabric such as a computer network and configured to create or modify an executable media package based upon input obtained from a plurality of contributing users (see e.g., Figure IA, 171, 172, ... 173n). The system provides an enterprise solution that can scale to any number of users. Distribution of the executable media package can be achieved using any communication and computational mechanism that enables users to obtain and view the contents of the executable media package (see e.g., Figure IA, 170). In at least one case the executable media package comprises data such as podcast data or other data obtainable via a subscription feed such as Really Simple Syndication (RSS) protocol. The podcast data is assembled from media data 176 and generated in one embodiment of the invention via transcoding process 174 that outputs at block 175 a generated podcast file with encoded chapter data and metadata. The generated podcast may include advertisement tracking

and incentivized click-thrus in one or more embodiments of the invention. The system is configured to accept media data that contains pretagged inventory where the pretagged inventory 180 is associated with a specific user attribute such as a demographic or geographic characteristic. An advertisement targeted at a specific demographic group is an example of pretagged inventory. The subscription feed is generated as part of the publication process at block 181 and generally summarizes the content of the media data. Hence the subscription feed is what provides users who wish to execute the executable media package with a way to subscribe to a specific set of executable media packages. Once an executable media package is assembled for subscription embodiments of the invention provide a mechanism for distributing the executable media package in order to achieve a business goal such as the desire to have viewers within a specific demographic access to the message contained therein. This distribution occurs via at block 182 where the executable media package is published along with the embedded metadata. Once published the executable media data is disseminated to requesting users (e.g., those subscribed to the feed) upon request. If updates to the executable media package are available 184 or if new media is published users can obtain the updated media 185 via the subscription feed process.

[0025] Although the invention is not limited to any specific user interface layout one or more embodiments of the invention make use of a series of interfaces configured to optimize the workflow process associated with generating executable media packages in cases where multiple users contribute to the creation of the executable media package. Because systems embodying the invention are able to accept input from multiple contributing users the system is well suited to become an integral component of any post-production workflow intended to allow multiple users to contribute to the creation of a set of executable media packages. For instance, software implementing aspects of the invention present a media collection interface configured to provide users with a way to identify data for inclusion into the executable media package, a chaptering interface for users to define navigation data and/or other metadata to be associated with a specific playback point, a media management interface for organizing and previewing media, and a publication interface for releasing the executable media packages to an audience. Each of these interfaces as well as other interfaces not expressly mentioned herein may be configured to provide additional functionality and examples of such functionality are described in further detail below. Readers are to take note however that such examples are not intended to be construed as

limiting or detract from the systems goal which is to make use of any interface or series of interfaces able to effectively collect content from multiple users for inclusion into an executable media package.

[0026] Systems embodying one or more aspects of the invention may be configured to implement various levels of user control. The general purpose of having such control in a multi-user environment is to provide a mechanism for production oversight prior to dissemination of the completed executable media package. This enables an administrator to control at the user level which users can access data and media data of a particular type. The user responsible for final review and publication of the executable media package may, for instance, differ from the users who create the content. In cases where access is restricted the restricted user is only given access to data within the users permission level.

[0027] System Components and Interfaces

[0028] Figure 1B is a high-level block diagram that illustrates the basic functionalities required to implement a solution that assembles data from multiple networked sources into media content for purposes of distributing said media content to a targeted audience of requesting users. To achieve such functionality the system contains an upload function (100) for uploading media data that is to be included into the executable media package. Audio data, video data, image data, text data and other forms of media content are some examples of the type of media data that can be uploaded for use. Generally media data can be obtained from anywhere on the local machine or the network that is accessible to the contributing user. A user may for instance use locally stored audio, images, or video data and/or use the local client computer to capture such data. For instance, in one embodiment of the invention media data is stored on the local system of the contributing user and uploaded to a remote system that is accessible to other contributing users. In this case the system uses a client/server topology where there are multiple clients configured to communicate with a scalable server. Users may also obtain media data from other computers accessible via the network and then upload the media data to the server. In one embodiment this client/server topology makes use of a thick client such as a Java™ Swing application communicating to a remote FTP server in the backend.

[0029] It is also feasible to have a single device function as both a client and server or to configure the system to make use of a peer-to-peer or other acceptable network topology.

Hence although the current invention is described in the exemplary form set forth herein using a client/server model it is not limited explicitly to use of a client/server model.

[0030] Figure 2 provides an example of the media collection interface configured to provide upload functionality in accordance with one or more embodiments of the invention. Referring now to Figure 2, media collection interface 200 contains a screen region having functionality that enables users to select media data via media selection implements 201-204. In the example depicted a command buttons provide users with a way to select audio 201, text 202, video 203 and images 204. Once the desired data is selected it is shown in the files to upload region 205 of the media collection interface. When the user has completed the selection files are uploaded to a centralized media data pool via the upload files command 206. The select images command may be utilized to select an advertisement source with a level of indirection that allows for dynamic advertisement inclusion. In addition, after upload, advertisement tracking components may be introduced into the podcast for metric generation functionality purposes. Alternatively, during the publish process, advertisement tracking components may be installed into the podcast based on any advertisement components that exist in the podcast whether direct or through a level of indirection.

[0031] Uploaded files are accessible for other users to access and take action upon. Chaptering interface 300 which is depicted at Figure 3 provides users with a tool for inserting navigation data and other metadata such as hyperlink data and images. Chaptering interface 300 is where users provide the inputs utilized by the assembly function 102 of Figure IB. The general purpose of assembly function is to generate executable media packages and define metadata to be associated with a specific package. Figure 1C provides a more detailed illustration of the methodology implemented via chaptering interface 300. At step 150 media data is obtained for playback. In the embodiment of the invention depicted in Figure 3, the "choose audio file" 301 command initiates a mechanism for obtaining media data that takes the form of audio data.

[0032] At step 151 media data playback occurs. Once playback begins users can perform various chaptering functions 302, 303, 304 to associate chaptering information and metadata with a particular media data timecode. The set first chapter function 302 enables the user to define where in time the first chapter begins (start time), the duration of the chapter, and also provide other metadata such as a chapter title, image data, and hyperlink information such as a URL and

URL title. The create chapter function 303 defines the start time, duration and associated metadata for additional chapters. These chaptering functions are how the system obtains chaptering input and metadata associated with a specific time code (e.g., step 152). Users can also associate default metadata such as images 305 and other information such as URL 306 and title 307 information with the media data at a non-specific time code. The delete chapter function removes a chapter and if necessary the system renumbers the existing chapters to maintain proper numbering. In cases where chaptering input is placed before an existing chapter mark (e.g., 153), systems embodying one or more aspects of the invention resort and renumber the chaptering data to maintain sequential organization (e.g., 154). If playback is complete and the user has finished setting the chapter data and the metadata, the chaptering functionality is complete and the user may cause the system to transcode the media data into an assembled executable media package. In cases where the executable media data is a podcast this is accomplished by initiating a create executable media package / podcast media file command 308 which in turn causes the system to assemble a executable media package / podcast enabled or compliant media file. To accomplish this the system executes at the command level functions to input the media data and a Data Transfer Object (DTO) containing XML based chapter information. The DTO functions as a conduit for client/server communications in that the data used by the client is put into a serializable object and sent to the client for handling. Hence the DTO is used to provide the client with the information needed to display the data. The client performs HTTP posts to communicate with the server. Referring now back to Figure 1B media management function 104 contains functionality for enabling a plurality of users to access and manage the media data pool containing data chaptered using the chaptering interface discussed above. An example of the media management interface is depicted in Figure 4. Screen region 401 provides a list of chaptered executable media packages that are available in an assigned workspace. In the example depicted screen region 401 provides a list of audio files wrapped within an MPEG-4/7 container format and having associated metadata such as that defined via chaptering interface 300. Using media management interface 400 users may select the chaptered files using a pointing device and execute a download operation to the network location identified with destination command 403 via command 402. Having this functionality gives users a mechanism to preview and test chaptered executable media packages on one or more local systems before they are released for publication.

[0033] Once the executable media packages are in final form users with the proper level of access make use of a publishing interface to submit the executable media data to a subscription feed. Various types of subscription and non-subscription interfaces are within the scope of the invention. Figure 5 depicts an example of a publishing interface for submitting executable media packages to a subscription feed. Publishing interface 500 contains various input regions for obtaining metadata to be associated with the publication. Once an executable media file is identified for publication (e.g., 501) information such as category information 502, a title 503, a description 504 about the contents of the package to be published and other extensions to the metadata such as link information, layout information or client specific information used by the client program that will be used to run the executable media package. Preview 505 provides users with a way to review the executable media package before submitting it for publication.

[0034] In one embodiment of the invention pre and/or post-tagging of the executable media file occurs. For instance in cases where the executable media file is an MPEG-4/7 video/audio file published data can be embedded in the MPEG file header metadata. Once embedded, the pre-embedded information enables batch publishing of episodic/syndicated content for distribution. This feature allows for the instant publishing of RSS metadata associated with an attachment as it is posted to multiple channels for distribution.

a. Name	I Lucy Lucy
b. Episode #	101
c. Description	Lucy takes a job at a candy factory
d. Season	Season 5
e. Content Advisory	TBD
f. Release Date	1958 or 01/01/1958
g. Copyright	©1966 Desilu Productions
h. Comments	Poor film to Video transfers. Scratches + Audio pops.

[0035] The publishing user may optionally associate demographic information with an executable media file and/or target a specific channel or even an episode on a channel at an intended target user. This enables the publisher to offer content based on a specific user profile or demographic. Profile information is typically obtained from users during a sign-up process but can also result from recording user behavior over time. When a user who has the targeted profile logs-in to the system access is granted to the published executable media package associated with their profile.

[0036] In one embodiment of the invention the information associated with a specific executable media package (or a channel containing multiple media packages) is indexed and can be used for searching (e.g., Figure IA, 186). The system makes use of manually entered or automatically generated index data associated with any executable media packages and thereby enables users to perform searches that are semantics based, content based, or based on metadata. Using this indexing approach it is possible for users to search based on people, places, events, scenes, color, texture, patterns, sound, genre, title, author, dates or any other information that might identify one or more executable media packages.

[0037] In one embodiment of the invention the system is configured so that voice, image, and speech-to-text software auto-catalogue and index content. This process automatically creates metadata that in turn enables text-based searches for specific multimedia content. The Metadata created can be added to the RSS description or other custom field. Also, if needed the software can capture the video (or audio) frame/time code/subject and create a .jpg snapshot frame to use for purposes of having a visual reference cue 602. For example a "Tom Cuise + Katie Holmes" search entered into a search interface 601 would query the database and find a .jpg of the frame w/ time code stamp (see e.g., Figure 6, 600). The search could pull up multiple reference jpg's from a library and display them as a grid collection along with text information. This service can be used by the Content creator and end/or users to search for content. Advertisers can use the search service to find contextual content that matches their product campaign profile.

[0038] A search within an episodic syndication post could appear as follows:

Channel: Inside Edition Season 10
 Episode: 101
 Date: 03/19/2006
 Description: Oscar ceremonies gala event.

TC: 01:21 :21 Tom + Katie Holmes
 TC: 01:21:25 Michael Douglas + Katherine Zeta Jones (Face)
 TC: 01:25:13 Lindsey Lohan (Face)
 TC: 01:29:22 Lindsey Lohan (Speech)

[0039] One or more embodiments of the invention are configured to implement channel or episode specific advertising (see e.g., Figure 7). For instance, multiple channels of the same content based on demographics (age, Income, interest). Ads can be changed on a daily, weekly, monthly basis or at other periodic intervals as desired. Advertisements can be submitted as

follows in audio, video or enhanced audio or video form. Advertisers may optionally embed hyperlinks into the enhanced media (e.g., enhanced podcasts) and optionally implement trackable unique affiliate marketing programs. Click-thrus to a hyperlink can then be measured and used to access effectiveness, obtain user provided data and/or extend ad viewing time. Using layering for example in a podcast file allows hidden functionality to be bundled into the podcast which may be utilized to track advertisement viewing. Interactive Flash, pictures, audio, advertisements and scripts may be embedded into a podcast. Once embedded, a script may be utilized to record the IP address and time and date of the tracked event. For example, a MOV file embedded into a low layer that is really a script, generates an error which allows the script to record the IP address and time and date of the execution of the script. Any other method of determining the time that an advertisement is viewed by a user is in keeping with the spirit of the invention. By timestamping events, whether through hidden embedded functionality or through any other method, the verification of advertisement downloading and viewing is achieved. The information related to the advertisement viewing may be tracked and reported to an advertiser or other content provider. In addition, specific targeting of advertisements may occur when a user opts into a profiling form, or when the user purchases something for example. As shown in Figure 7, user B may have provided profile information that allows for the targeting of advertisements AD 2 and AD 3 to user B.

[0040] Referring again to Figure 7 channels 1-A, 1-B or 1-C are all different variations of a channel having the same basic episodes but having variable advertising content (e.g., AD 1, AD 2 and AD3). The subset of the channel that is provided to the user in the form of an executable media package depends on profile or login information of the user or on any other characteristic associated with the user, time or any other characteristic. When user A subscribes to the system the user might receive Channel 1-A, user B could receive Channel 1-B and a user C might receive Channel 1-C. The difference between what channel users receive is based on information provided by the user or information learned about the user through recording the users usage habits. By tracking the viewing of advertisements, the system can provide metrics to advertisers that indicate the level of viewing of advertisements that is occurring. In this manner, popular podcasts may demand higher rates of compensation from advertisers.

[0041] In addition to tracking podcast related advertisement viewing, one or more embodiments of the invention enable incentivized click-thru functionality. For example, by presenting an advertisement embedded into a layered podcast, e.g., podcast for channel IA and obtaining a click-thru, incentivized click-thru functionality is achieved when redirecting to a merchant who delivers free media content. In order to incentivize the click-thru, the system provides a free product, for example a free media file or song. When the user clicks-thru, the free song is provided. When the cost of the song is less than the price of the click-thru, then a profit may be generated. As shown in Figure 7, if a user B clicks on AD3, then the user may be presented with a website associated with AD3 and when the system records the click-thru, the advertiser pays the merchant for redirecting the user. Using these embodiments of the system allows for the generation of income when content producers or advertisement providers pay a merchant for click-thrus. In certain countries where media is generally not paid for, any free giveaway of media that results in payment to the content creators provides an incentive for content creators to enter that market. For example, in China a click-thru cost of 63 cents with a cost per song of under 8 cents provides a large incentive for using an embodiment of the invention compared to the United States market where a click-thru cost of 1 dollar with a cost per song of 60 cents providing a lower, yet positive incentive. Since certain countries allow for exclusive rights, for example for podcast rights, a percentage of click-thru revenue may be contracted to a granting authority for the exclusive license to provide the system enabled herein. Ad content may be auctioned out allowing for different media buyers to provide ads that dynamically change based on any variable, such as the time of viewing, after a number of ads has been displayed or based on the profile of the user viewing the ad or any other characteristic associated with the user, location of user, time, or any other variable.

[0042] Figure 8 illustrates an executable media package in the form of a podcast that is configured to save click-thru information and provide a user interface element that results in a free product giveaway if the user clicks-thru the link. Executable media package is shown visually in viewer 801. In this example, the executable media package or podcast is depicted with link 802 that results in the display of web store 850 and link 803 that results in a download of a free product, or display of a web site or web store where the product is free or is marked as free by executing link 803. Executable media package play position 804 shows that the

executable media package is approximately 2/3 through playing. At this point in the podcast, link 802 or 803 may be displayed so that the links are available only at that portion of the podcast. The media played in viewer 801 may include advertisements for example that are viewed before links 802 and 803 are displayed to ensure that clicks through these links signify that a user has been shown an advertisement. Alternatively or in combination, hidden scripts may be utilized to ensure that advertisements have been seen by a user. When a user has viewed an advertisement, the IP address, advertisement ID and timestamp of viewing may be saved on any computer for further processing such as billing of an advertiser. For free product giveaways, a content producer may pay the merchant for the product and yield a net profit by charging more for a click-thru to an advertiser. For example, by charging an advertiser more money for clicking through link 803 and the cost of a product, e.g., song in this example, a net profit is produced. The merchant may also pay the content producer for directing users at their website or web store as well. In addition, the advertiser generally pays a click-thru and page view fee to the content producer that has generated the podcast.

[0043] The workflow process for handling targeted advertising involves the content provider uploading executable media packages and submitting a plurality of advertising content (e.g., AD1, AD2, AD3) via a publishing process. One or more embodiments of the invention add the channel to a database of inventory from which advertisers can search and select a channel from the available inventory within which to place advertisements. The system creates an ad inventory database by searching for "black" frames (or silent audio) longer than a threshold time period that is typically less than the time period users can easily perceive or generate frames from videotape archives. Time code logs for the advertisements are included in the database and ads can either be manually or automatically submitted at chosen time intervals (e.g., every 15 minutes, bookend, (Beginning/Middle/End)). Advertisers can then search and preview content based on entered search criteria. Some examples of search queries that might be initiated by a user who wishes to place advertisements follow:

USER	SCENARIO	SEARCH
Movie Studio	MI3 campaign	all references to Tom Cruise
Pet Company	Pet food	Dogs and Bark Sounds
Geritol	Senior Product	I Love Lucy + Vivian Vance

[0044] After performing searches, advertisers or media buyers can drill deeper to obtain channel/episode information, determine inventory availability, and ascertain the cost of placing the advertisement. Once an advertisement is selected it can then be placed into the appropriate content as needed. An administration console allows advertisers to track and monitor an ad campaign and make changes to the campaign as desired.

[0045] Design Details

[0046] Although there are various ways to implement the general functionality and interfaces described herein, a more specific set of details is provided to give an example of the modules and basic functionality used in the context of a client/server embodiment of the invention.

[0047] On the client side one or more embodiments of the invention makes use of the following modules:

[0048] HTTP connection module comprises a collection of classes working together to make http requests and receive http responses from the backend server of the application. The HTTP connection module may also pass the user login data and a Jsession Id generated by the application server for all communications with the client. The connection module will store this data and use this for all communication with the backend server (Server Side).

[0049] Swing UI module contain the program code used to implement the various interfaces described throughout this document and in one embodiment of the invention comprises the various Swing components and widgets to be used in the client. An observer framework is used for event handling.

[0050] Session Data Module is a storage container containing the server session data (e.g., userinfo and JSession Id etc). This data may, for instance, be contained within the HTTP connection module and written to a log locally for debugging purposes.

[0051] Data Transfer Object (DTO) is a storage container configured to act as a conduit for the server to communicate with the client. In one embodiment of the invention the data used by the client is put into this serializeable object, sent to the client and used by the client for display data. The client performs HTTP posts to communicate with the server.

[0052] On the server side one or more embodiments of the invention is implemented as a Struts application which follows the MVC (Model View Controller) architecture and has Multi-part form support used to post file data. The web application use these primary URL mappings to service requests.

[0053] An Upload Files request receives multipart data from the client and writes it into the account folder as a file. The data that is uploaded in this context is the original source data that may or may not be included in the executable media packages when they are finished. Duplication files names are also handled.

[0054] A File Download service is used in one embodiment of the invention by the create Podcast command to play an audio file or in the more generalized case to play any media data. The media data is downloaded using HTTP and played in the Java QuickTime Player in a way that outputs the current time code for purposes of creating chapter data. The chapter data is written to a serializeable DTO and posted once the Create Podcast Button is pressed.

[0055] Login and Log Off functionality utilizes the user table. This module will be used to manage and control user access. Primarily maintaining sessions when users are logged on or off of the system. For instance, one implementation of the invention makes use of a database to persist user data. This is helpful in a multi-user context because users can be given access levels and an administrator can exercise control over each users. Although not set forth in the table below the database can also be utilize log tables to log the user activity. On the viewer side of things a user login is used to associate the user with a particular profile and thereby enable advertisers to target specific content at the user.

[0056] Not by way of limitation and included herein only for purposes of example is a sample users table.

USERS Table		
Column Name	Column Type	Column Description
USERID	Big Int Not Null – Primary Key	Unique ID of User
USERNAME	Varchar2	User’s Name
PASSWORD	Varchar2	User’s Password
ACCOUNTID	Big Int	Account Id

SECURITYLEVEL	Integer	Used to define user's access privileges
VALIDFROM	Date	Date when account is enabled
VALIDTO	Date	Date when account is disabled
CREATEDDATE	Timestamp	Date when account was created
CREATEDBY	Varchar2	User who created the account
UPDATEDDATE	Timestamp	Timestamp of when the row was last updated
UPDATEDBY	Varchar2	User who updated the row

[0057] The create podcast service is called when the Create Podcast button is pressed (see e.g., Figure 3). In one example, input is the file names and the DTO containing the XML based chapter info is processed at a command level to create the Podcast.

[0058] The Data Access Object (DAO) module is used to allow the application to access the Database and uses a JDBC to connect to the database. Connection pooling provides a way to increase availability.

[0059] RSS feed. In one or more embodiments of the invention the basic information defined via publishing interface 500 is submitted in the form of an RSS feed containing XML data. Although the invention is not limited solely to the use of RSS feeds, it is useful for purposes of illustrating how publication occurs to discuss the basic structure of an RSS feed. When submitting a feed the information that is to be posted with the feed is submitted as XML data. The outer container is <rss> tag that encloses a <channel>. The channel contains elements or tags that collectively define feed properties via the accompanying metadata. Following the channel elements are items that define the actual feed content. Each item within the XML data may define a unique entry, and a feed can contain multiple items. An <item> element, for instance might define a headline, image, or audio file. Closing </channel> and </rss> tags will follow the items in order to maintain proper syntax. Variations of the example show below will occur depending upon the data provided via the publishing interface and hence one of skill in the

art will recognize that the example set forth below is a generalized example that is none specific to an actual publication case.

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>

  <!-- channel metadata -->

  <title>My Executable Media Data</title>
  <link>http://A URL/feed.rss</link>
  <description>Headlines and important news</description>

  <!-- channel content -->
  <item>
  <!-- item content -->

    <title>Intro To Developing Executable Media Packages</title>
    <description>This article provides an overview of the methodology used to develop
executable media packages. The enclosure references the podcast.</description>
    <enclosure url="http://[URL]/Assembled_Package.m4b" length="123456"
type="audio/mpeg" />
  </item>
</channel>
</rss>
```

[0060] Module Specific Functions

[0061] The Log In to Client module provides the functionality that enables users to login to the system. Since one or more embodiments of the invention make use of a thick client for the Application front end, the login dialog can be presented once the user clicks on the application. Once the user enters the username and the password and clicks on logon, the Event handler framework on the client checks the triggered event (login) and passes over the input widget info over to the LoginAction Class. This class is programmed to perform validation of the input data

(username/ password) and then create a Data Transfer Object (DTO) with the data and pass it to the HTTP connection Module which then calls the Login service on the Application server. The application server passes this request to the Login Action class on the server side. This server side Login Action class then authenticates the user and sends back to the user the userinfo and the Jsessionid in a DTO. The user is logged in and is shown the main page in the application client. If the user entered data is invalid the user receives an error message.

[0062] The Main Menu UI module contains functions for rendering the interface the user views once logged in. The HTTP connection module obtains the response from the server about the login attempt and if it is successful places the userinfo data and the Jsession id in the SessionInfo module and calls the Main page UI widget. In one implementation this widget contain numerous buttons with labels for functions such as: User can upload audio files (e.g., *.m4a and other audio file types) User can upload image files (e.g., *.gif, jpg, .png and other image file types), (Transcoding will support multiple file formats). Create Podcast File, Create RSS Feed from existing Podcast enabled audio files.

[0063] The Upload Audio Files module provides the user with an upload media data option. The event handler framework present the user the LOAD media widget (e.g., audio or video data) which contains a button to show the file chooser. The user can use the filechooser to choose a media file. The user may select the file and then click on upload button. The event handler framework posts this multipart data along with the session info to the Application server. The application server then calls the UploadAudio Action class which parses the multipart data and writes it to the directory of the account associated with the user. While this is happening the user is optionally shown a dummy progress bar. The file is uploaded and the user get a file upload finished dialog with option to upload more. If the user selects yes then the event handler framework shows the same Audio File upload widget again.

[0064] The Upload Image Files module is executed when the user initiates a command to obtain an image. The event handler framework responds by presenting the user the LOAD Image widget. This widget contains a button to show the file chooser. The user can use the filechooser to choose an image file, select the file and then click on upload button. The event handler framework posts this multipart data along with the session info to the Application server. The application server calls the UploadImage Action class that parses the multipart data and write it

to the image directory of the account associated with the user. The file is uploaded and the user will get a file upload finished dialog with option to upload more. If the user selects yes then the event handler framework shows the same Image File upload widget again.

[0065] When the Create Podcast file module is executed the event handler framework presents the user the Create Podcast widget. This widget contains a button to choose the audio file. When the user clicks on the button the event handler framework sends this request to the HTTP connection module. The connection module calls the Application server with this request. The application server then passes the request to the ShowServerAudio files class. This class will read the names of all the files in the account directory associated with the user and puts these filenames in a DTO and sends the response back to the client. The client passes this DTO response to the filechooser widget and shows it the user. The user may then select a file and click the view button. When the user clicks on the button the event handler framework sends this request to the HTTP connection module. The connection module calls the Application server which in turn passes the request to the DownloadServerAudio files class. This action class calls the download module which sends back the client the file chosen by the user. The HTTP connection module can download this file to a temporary location and start playing it in a java based QuickTime player widget. The Create Podcast file module may be utilized to introduce metric generation of advertisement viewing. In one or more embodiments, the metric generation may comprise inserting an event time stamp module that is configured to time stamp the viewing start and end times for an advertisement. By sending the IP address, advertisement ID and start and stop times to a computer configured to store advertisement information, advertisement metrics may thus be stored.

[0066] The user can click on the create chapter button. When the user clicks on the button the event handler framework creates an editable chapter name with the time code on the player. Users can create multiple chapters. The data is stored in a Data Object. On the UI, each of the chapters has a button. The user can click on the button to associate the (image file) gif file to the chapter. This is done in a similar way to the previously explained method (View Audio File). Once the user is satisfied with the editing, the user can click on the create PODCAST (Podcast File) button. This button passes the chapter data object to the HTTP connection module which passes the request to the application server. The application server passes the request to the

CreatePodcast Action class which processes the data and passes it to a UNIX command line utility that convert the audio to the Podcast file. The Podcast file is stored in an output directory.

[0067] Create RSS Feed. The user will click on the button next to the label containing the Create RSS feed option. The event handler framework will present the user the Create RSS feed widget. This widget will contain a button to choose the Podcast file. When the user clicks on the button the event handler framework will send this request to the HTTP connection module. The connection module will call the Application server with this request.

[0068] The application server passes the request to the ShowPodcastFiles action class. This class reads the names of the Podcast files in the account directory associated with the user, places these filenames in a DTO and sends the response back to the client. The client passes this DTO response to the filechooser widget and presents it to the user. The user may then select a file and click the create RSS feed button. When the user clicks on the button the event handler framework send a request to the HTTP connection module. The connection module calls the Application server with this request. The application server then passes the request to the CreateRSSFeed action class that calls the RSS Feed module. This module passes the input request parameters to the RSS feed command line utility which generates an RSS feed out of the input.

[0069] Media File Player module. The player used to play the media data (e.g., audio data) will need to play, scroll, stop playback, and display timecode. Q4J player (Java QuickTime player) is an example of a player that serves this purpose.

[0070] The Logout module provides the functionality needed to log the user off the system. Once the user clicks on logoff, the Event handler framework on the client checks the triggered event (logoff) and passes over the input widget info over to the LogOffAction Class. This class invalidate the user session and send a response to the client about successful logout. Data in the Session module of the client is typically be removed.

[0071] These details set for in the form of the examples contained herein comprise a system and method for assembling data from multiple network sources into executable media packages configured for targeted dissemination to an audience of requesting viewers has been described. While the invention herein disclosed has been described by means of specific embodiments and applications thereof, numerous modifications and variations could be made thereto by those skilled in the art without departing from the scope of the invention set forth in the claims.

CLAIMS

What is claimed is:

1. A computer program product for assembling data from multiple networked sources into media content for purposes of distributing said media content to a targeted audience, said computer program product comprising:
computer readable program code embodied in a computer usable memory medium where said computer readable program code is configured to:
present a media collection interface configured to provide a plurality of contributors with a mechanism for identifying data for inclusion into an executable media package;
obtain input to identify said data from said plurality of contributors via said data collection interface where said input identifies a storage location for said data;
present a navigation interface configured to provide said plurality of contributors with a mechanism for defining navigation data to associate with said data;
obtain input to define said navigation data from at least one of said plurality of contributors via said navigation interface;
assemble said data and said navigation data into said executable media package;
generate a subscription feed comprising a content summary of said executable media package where said subscription feed comprises links to said executable media package; and
publish said executable media package for download by users subscribed to said subscription feed.
2. The computer program product of claim 1 wherein said media collection interface provides said plurality of contributors with access to said data where said data comprises audio, video or text data.
3. The computer program product of claim 1 wherein said computer program product further comprises:
save advertisement viewing information associated with said executable media package;

4. The computer program product of claim 1 wherein said computer program product further comprises:

provide a tree product or service to a user that has clicked-thru an advertisement.

5. The computer program product of claim 1 wherein said media collection interface provides said plurality of contributors with access to said data where said data comprises image data.

6. The computer program product of claim 1 wherein said computer readable program code is further configured to upload said data from a local file system executing said computer program product to a repository of data separable from said local file system.

7. The computer program product of claim 6 wherein said computer readable program code is further configured to:

download said data from said repository for playback with a time code at said local file system;

playback said data to at least one contributing user with said time code data for purposes of obtaining said navigation data from said at least one contributing user.

8. The computer program product of claim 7 wherein said navigation data comprises metadata associated with said data.

9. The computer program product of claim 8 where said metadata comprises link information associated with said data in said executable media package.

10. The computer program product of claim 8 wherein said metadata comprises image information associated with said time code.

11. The computer program product of claim 8 where said metadata comprises link information associated with insert data used to update to said executable media package.

12. The computer program product of claim 11 where different version of said executable media package are assembled and made available to users based on user profile information.

13. The computer program product of claim 11 where contents of said executable media package are modifiable by advertisers via an ad administration interface.

14. The computer program product of claim 11 where advertising data is variable in accordance with demographic information of said viewing user.

15. The computer program, product of claim 14 where executable media data is indexed to enable searching.

16. The computer program product of claim 14 where said advertising data is associated with an executable media file based on search criteria entered by an advertiser.

17. In a computer system a method for assembling data from multiple networked sources into media content for purposes of distributing said media content to a targeted audience of requesting users comprising:

presenting a media collection interface configured to provide a plurality of contributors with a mechanism for identifying data for inclusion into an executable media package;

obtaining input to identify said data from said plurality of contributors via said data collection interface where said input identifies a storage location for said data;

presenting a navigation interface configured to provide said plurality of contributors with a mechanism for defining navigation data to associate with said data;

obtaining input to define said navigation data from at least one of said plurality of contributors via said navigation interface;

assembling said data and said navigation data into said executable media package;

generating a subscription feed comprising a content summary of said executable media package where said subscription feed comprises links to said executable media package; and

publishing said executable media package for download by users subscribed to said subscription feed.

18. A system for assembling data from multiple networked sources into media content for purposes of distributing said media content to a targeted audience of requesting users comprising:

a means for presenting a media collection interface configured to provide a plurality of contributors with a mechanism for identifying data for inclusion into an executable media package;

- a means for obtaining input to identify said data from said plurality of contributors via said data collection interface where said input identifies a storage location for said data;
- a means for presenting a navigation interface configured to provide said plurality of contributors with a mechanism for defining navigation data to associate with said data;
- a means for obtaining input to define said navigation data from at least one of said plurality of contributors via said navigation interface;
- a means for assembling said data and said navigation data into said executable media package;
- a means for generating a subscription feed comprising a content summary of said executable media package where said subscription feed comprises links to said executable media package; and
- a means for publishing said executable media package for download by users subscribed to said subscription feed.

FIG. 1A

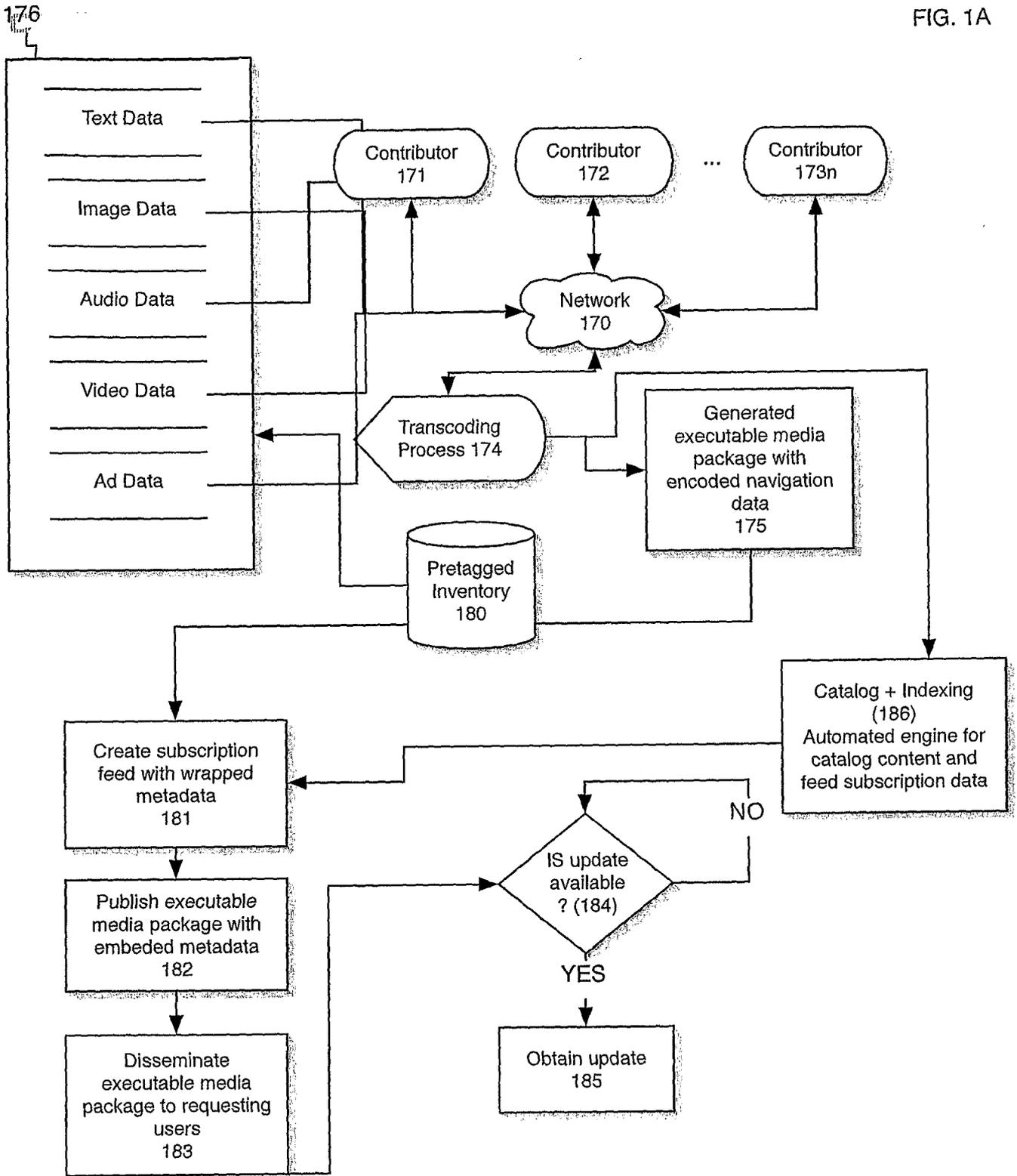
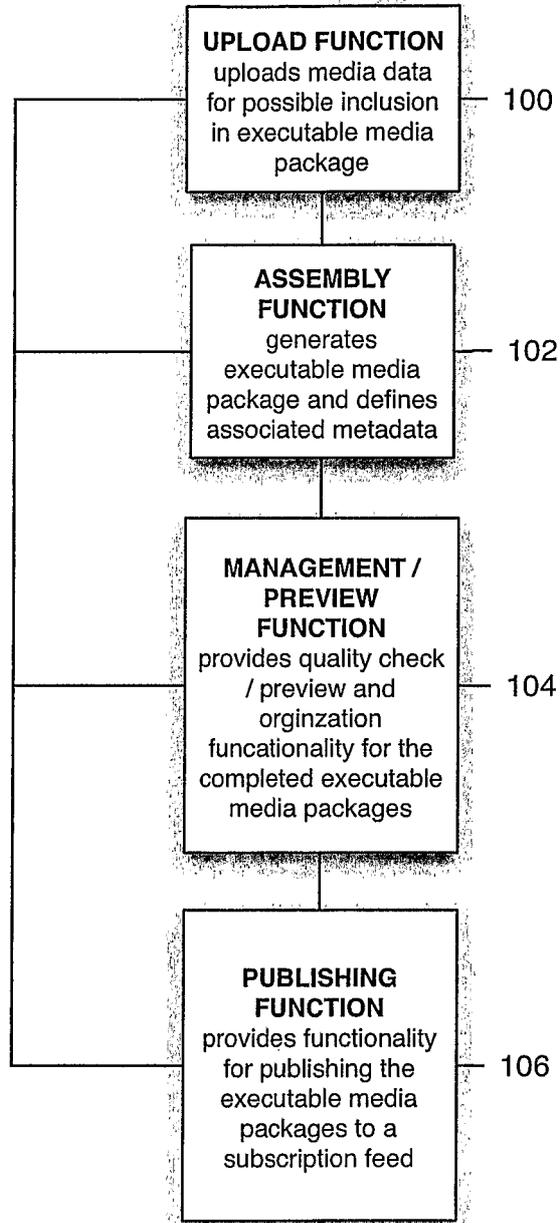


FIG. 1B



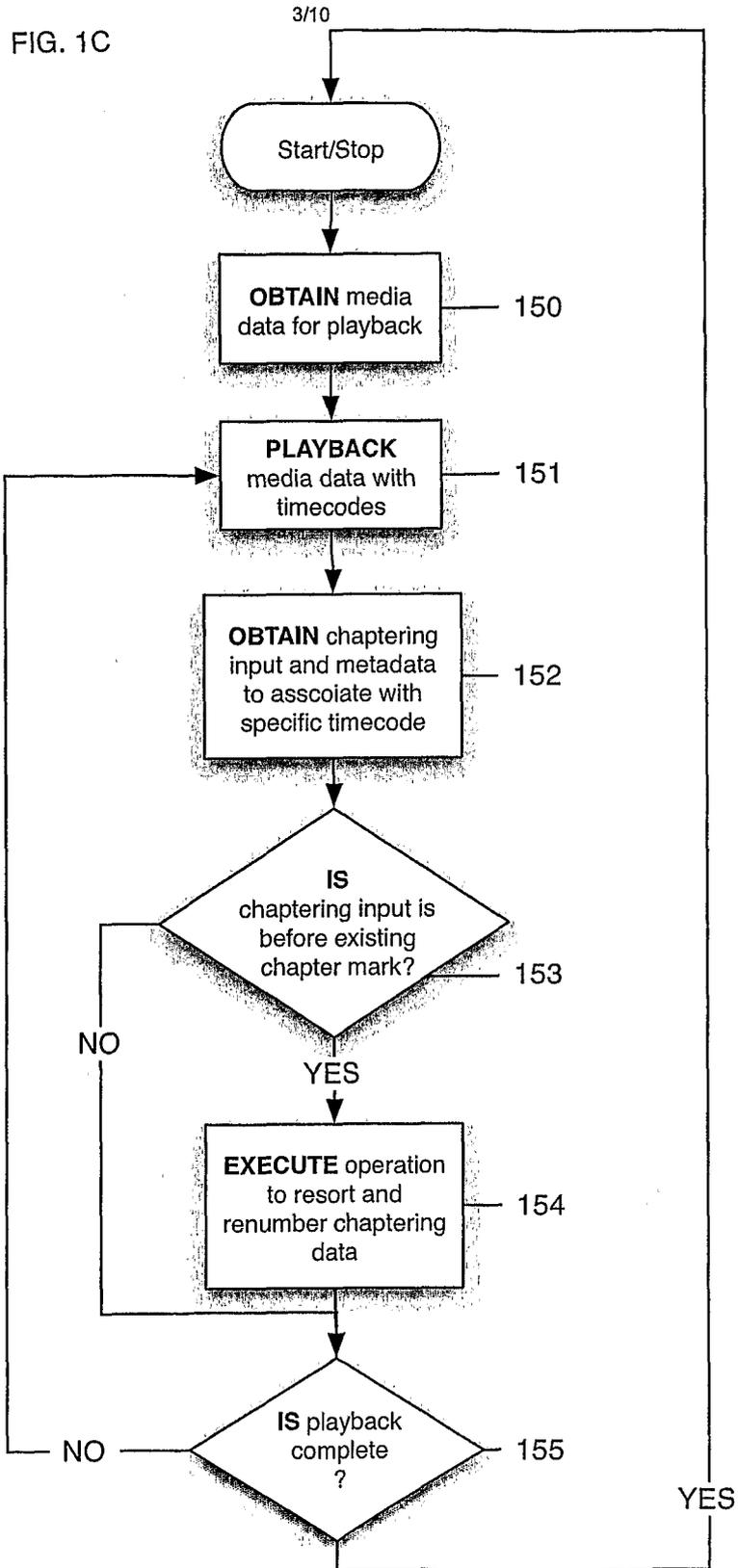


FIG. 2

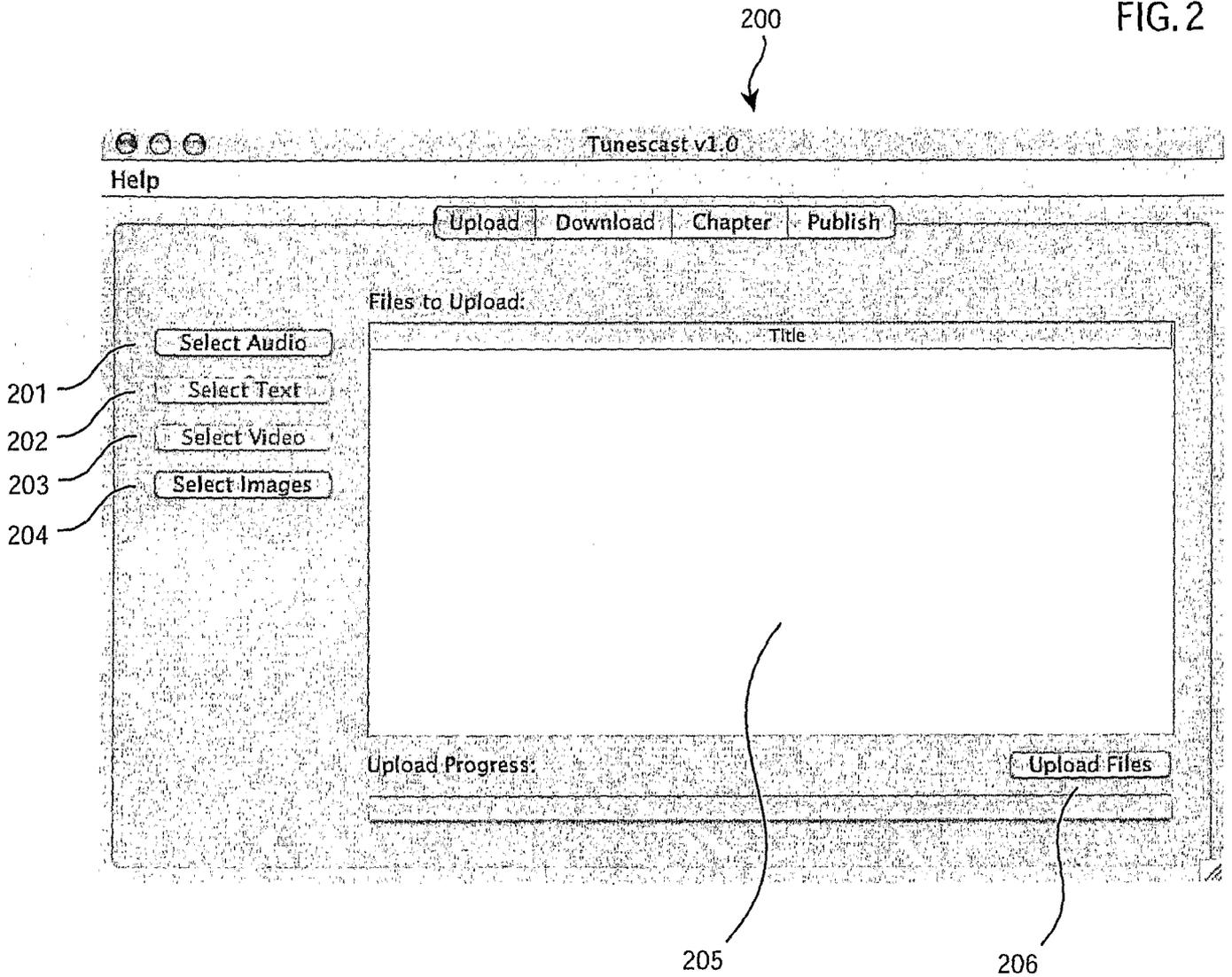


FIG. 3

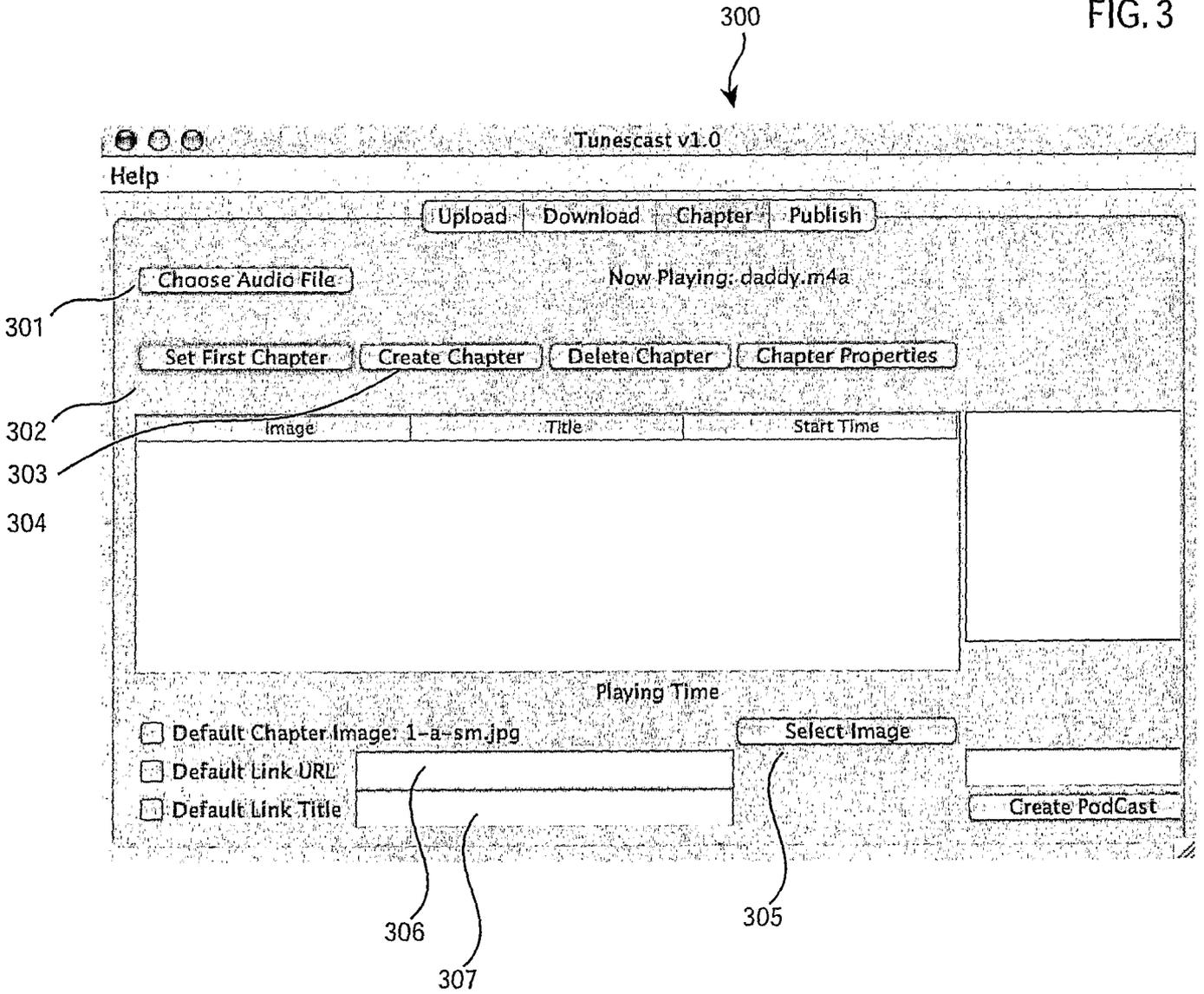
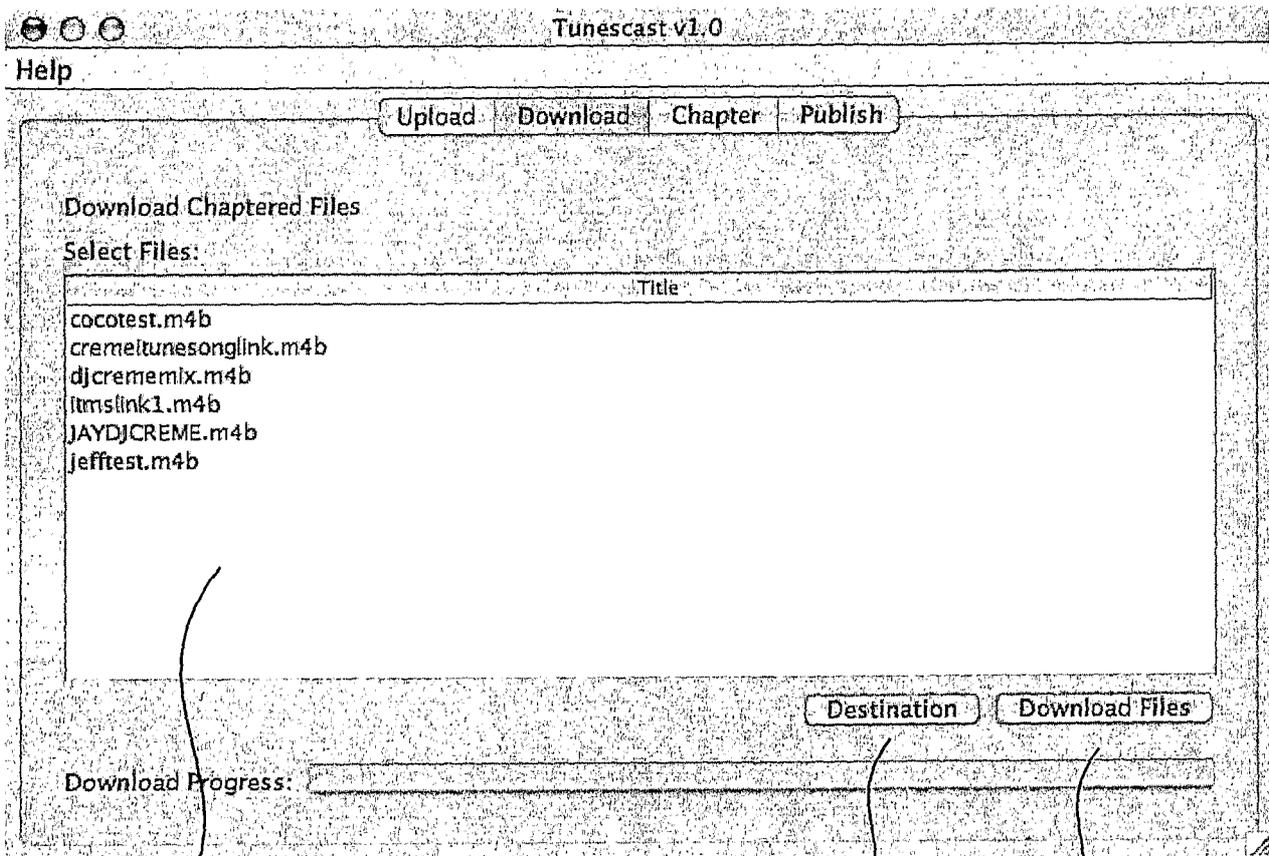


FIG. 4

400



401

403

402

FIG. 5

500

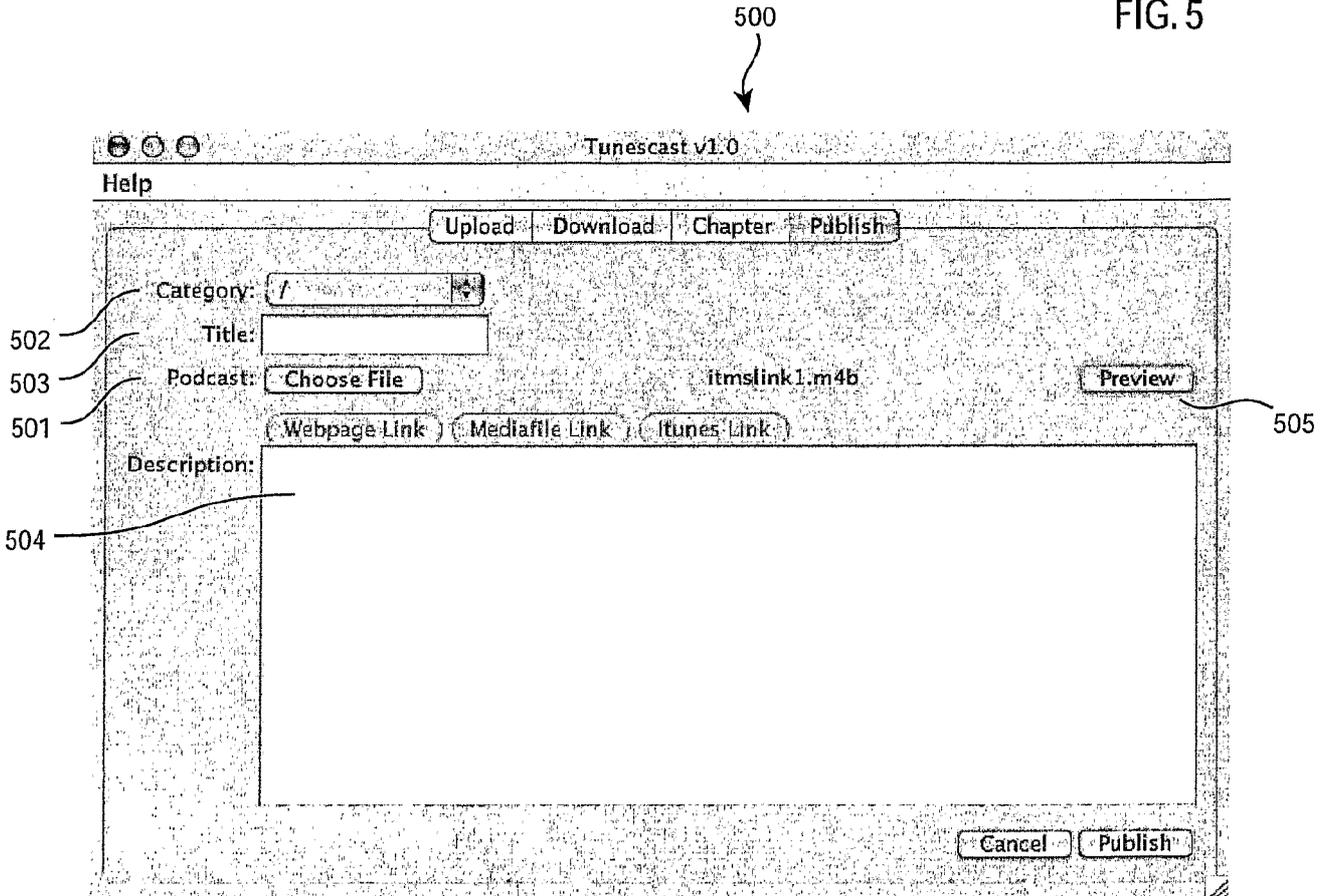
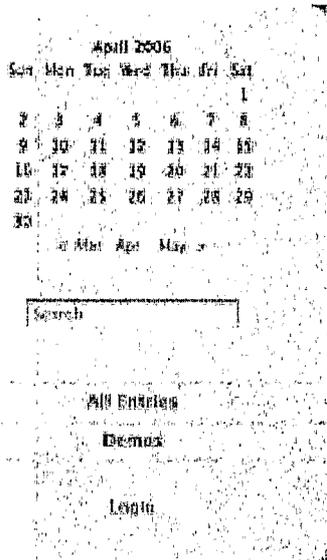
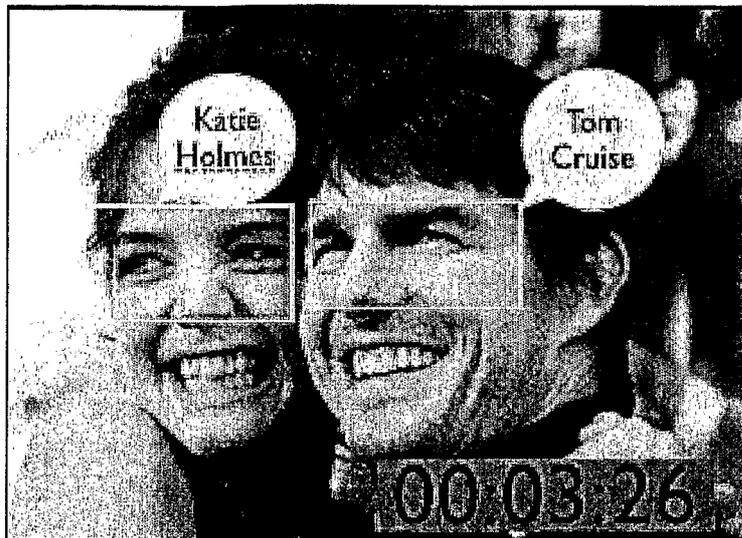


FIG. 6

Visual Reference Cue 602

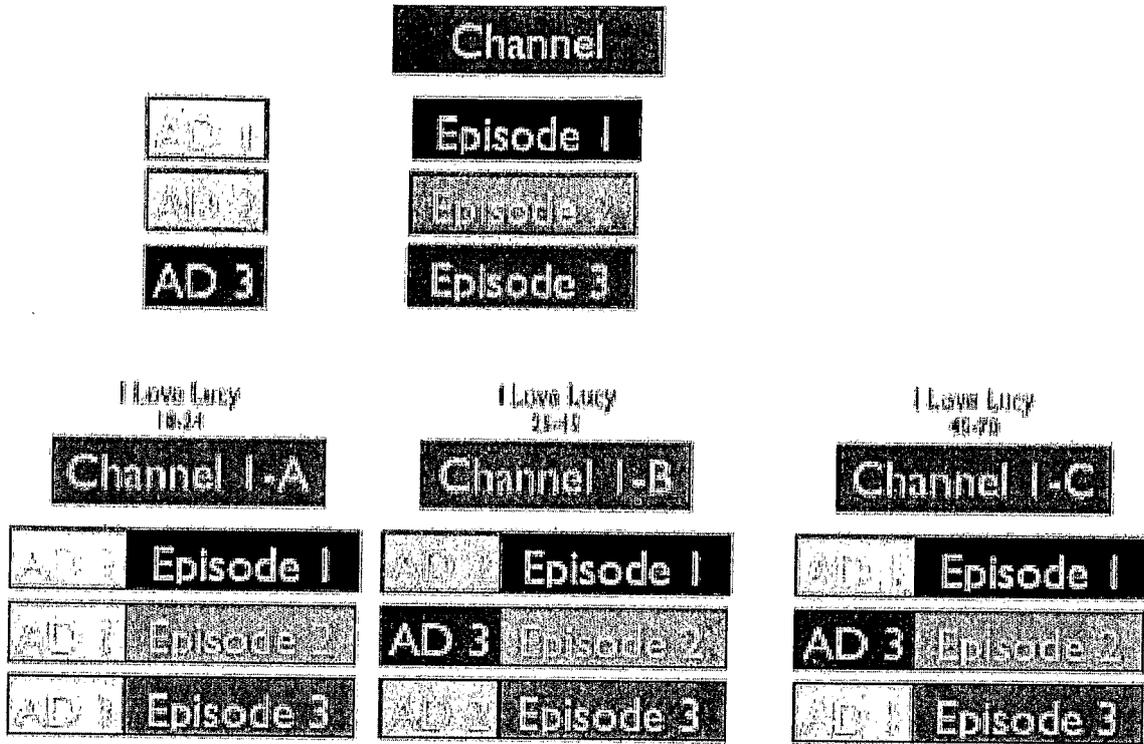


Search Interface 601



Time code 600

FIG. 7



INTERNATIONAL SEARCH REPORT

International application No.

PCT/US06/35989

A. CLASSIFICATION OF SUBJECT MATTER

IPC: G06F 11/00(2006.01)

USPC: 714/4;717/125;707/10;705/23,27

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)

U.S. : 714/4; 717/125; 707/10; 705/23,27

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 practicable, search terms used)
EAST

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 6,324,69 1 B1 [GAZDIK] 27 November 2001 [27. 11.2001], see entire document.	1-18
A	US 6,922,707 B2 [SHIMOJIMA et al] 26 July 2005 [26.07.2005], see entire document.	1-18
A	US 2002/0062320 A1 [SHIMOJIMA et al.] 23 May 2002 [23.05.2002], see entire document.	1-18
A,P	US 2007/0016530 A1 [STASI et al.] 18 January 2007 [18.01.2007], see entire document.	1-18
A,P	US 2006/019551 3 A1 [ROGERS et al.] 31 August 2006 [3 1.08.2006], see entire document.	1-18

 Further documents are listed in the continuation of Box C. See patent family annex.

* Special categories of cited documents	"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
"A" document defining the general state of the art which is not considered to be of particular relevance	"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
"E" earlier application or patent published on or after the international filing date	"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
"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)	"&"	document member of the same patent family
"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		

Date of the actual completion of the international search

11 May 2007 (11.05.2007)

Date of mailing of the international search report

25 JUN 2007'

Name and mailing address of the ISA/US

Mail Stop PCT, Attr ISA/US
Commissioner for Patents
PO Box 1450
Alexandria, Virginia 223 13-1450

Facsimile No. (57 1) 273-3201

Authorized officer

Jacqueline A. Whitfield
Dieu-Minh Le
Special Project Asst.
Telephone No. (571)272-3660