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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 January 2002 (10.01.2002)

PCT

(10) International Publication Number
WO 02/03644 A2

(51) International Patent Classification⁷: H04L 29/00

(21) International Application Number: PCT/US01/20133

(22) International Filing Date: 21 June 2001 (21.06.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/608,803 30 June 2000 (30.06.2000) US

(71) Applicant *(for all designated States except US)*: **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, CA 95052 (US).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **PENDKUR, Ramesh** [US/US]; 19766 NW Philips Road, Hillsboro, OR 97124 (US).

(74) Agent: **MALLIE, Michael, J.**; Blakely, Sokoloff, Taylor & Zafman, 7th Floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025 (US).

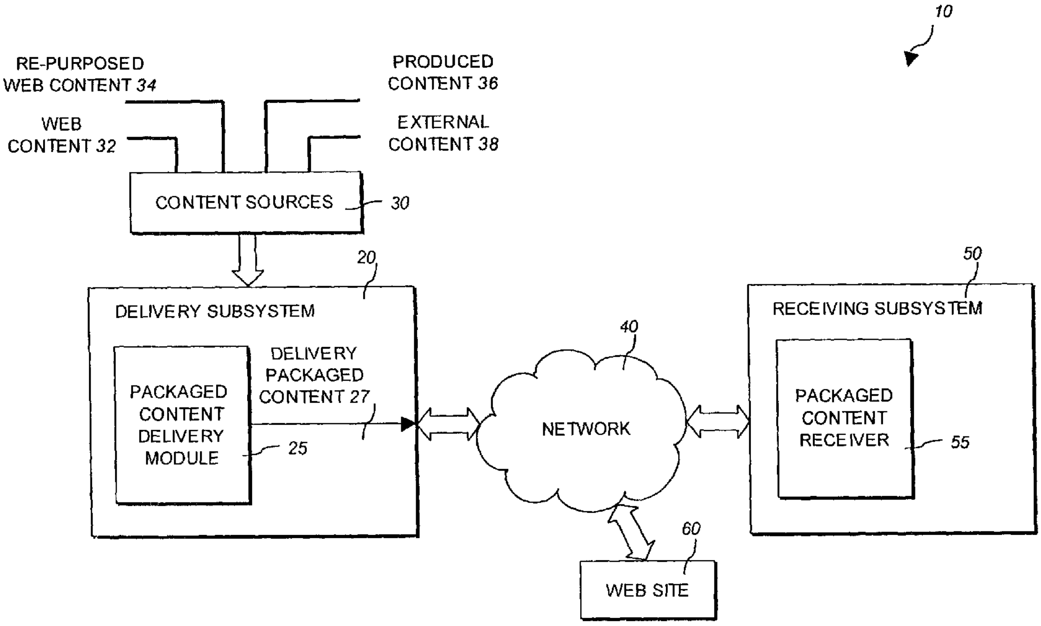
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU,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

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

**Published:**
— *without international search report and to be republished upon receipt of that report*

*[Continued on next page]*

# (54) Title: DELIVERING CONTENT BY ENCAPSULATING INTO PACKAGES WITH ASSOCIATED META-DATA



(57) **Abstract:** In one embodiment of the invention, a content is composed for delivery over a network to a subsystem. The content has a package containing a content. The package has package elements. The package and the package elements are tagged using meta-data. The meta-data are serialized into a file system compatible with the subsystem using a markup language.

# DELIVERING CONTENT BY ENCAPSULATING INTO PACKAGES WITH ASSOCIATED META-DATA

## BACKGROUND

### 1. Field of the Invention

This invention relates to networks. In particular, the invention relates to content delivery.

### 2. Description of Related Art

Information distribution over a network is becoming more and more popular. Users now can have access to thousands of network databases and content providers to retrieve news, articles, files, stock quotes, music files, video clips, etc. with ease. In many instances, content providers download data, news, and files to the subscribers automatically.

As demands for information access become higher and higher, it becomes economical to package the information contents for efficient delivery and discovery. However, the abundant supply of information has created several problems. The interests of users change over time, leading to difficulties in sorting out relevant information. The users' platforms or computer systems are constantly upgraded with new hardware and software, making presentation of information at the user's site less effective.

Existing packaging techniques merely bundle files into a .zip or .tar file. The content is usually compressed and integrated without any categorization or discrimination of information. These techniques provide no useful information about the content other than the names and sizes of the individual files.

Therefore, there is a need to have an efficient technique to package content for delivery.

## BRIEF DESCRIPTION OF THE DRAWINGS

The features and advantages of the present invention will become apparent from the following detailed description of the present invention in which:

Figure 1A is a diagram illustrating a system in which one embodiment of the invention can be practiced.

Figure 1B is a diagram illustrating a computer system according to one embodiment of the present invention.

Figure 2 is a diagram illustrating a packaged content delivery according to one embodiment of the invention.

Figure 3 is a diagram illustrating a package composer shown in Figure 2 according to one embodiment of the invention.

Figure 4 is a diagram illustrating a packaged content tagger shown in Figure 2 according to one embodiment of the invention.

Figure 5 is a diagram illustrating a packaged content serializer shown in Figure 2 according to one embodiment of the invention.

Figure 6 is a diagram illustrating a packaged content receiver shown in Figure 1 according to one embodiment of the invention.

## DESCRIPTION

The present invention is a technique to deliver content to a subsystem or node in a network. A package is composed for delivery to have package elements corresponding content of the packaged content. The package and the package elements are tagged using meta-data. The meta-data are then serialized and combined with the package elements into a serialized packaged content compatible with a receiving subsystem using a markup language.

The content is collected or gathered from several content sources. The content is encapsulated into packages with associated annotation and their respective meta-data information. The packages, once created, are used as units of distribution over the broadcast network.

A package is a logical grouping of media and other data files with associated meta-data about the content. A package is made up of package elements. Both the package and its package elements may have associated meta-data. Meta-data is defined as content or media descriptor tags that characterize the package elements and the package so that the corresponding content can be used in an intelligent manner within the broadcast network. The package elements may include HyperText Markup Language (HTML) pages, MP3 files, QuickTime movies, video clips, or any other suitable content. A package is independent of network and transmission mechanisms and policies. A package is represented in the file system with a unique identifier and package markup language data.

The meta-data assigned to the package and the package elements is unchanged in the system. In other words, the meta-data used at the sender and the receiver have the same meaning. The receiving subsystem may add additional meta-data, but the original meta-data as composed by the content originator with the delivered packaged content remains unchanged.

In the following description, for purposes of explanation, numerous details are set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that these specific details are not required in order to practice the present invention. In other instances, well-known electrical structures and circuits are shown in block diagram form in order not to obscure the present invention.

Figure 1A is a diagram illustrating a system 10 in which one embodiment of the invention can be practiced. The system 10 includes a delivery subsystem or node 20, a plurality of content sources 30, a network 40, a receiving subsystem or node 50, and a web site 60.

The delivery subsystem 20 provides the content to be delivered to other subsystems or nodes connected to the network 40. The delivery subsystem 20 may be a Web site, a content provider, an internet service provider (ISP), or any system that delivers, downloads, or broadcasts contents, packaged contents, data, and files over the network 40 to other subsystems or nodes. The delivery subsystem 20 may include a server, a computer system, or a platform with appropriate hardware and software to generate and prepare the packaged content to be delivered. One embodiment of the delivery subsystem is shown in Figure 1B. The delivery subsystem 20 includes a packaged content delivery module 25 to delivers a packaged content 27 containing the content to be delivered.

The content sources 30 include the sources that provide the content of the packaged content to be delivered by the delivery subsystem 20. These sources may include Web content 32, re-purposed Web content 34, produced content 36, and external content 38. The Web content 32 may include Web pages and files retrieved from a Web site such as the Web site 60. The re-purposed Web content 34 may include previously retrieved Web pages and files. The produced content 36 may include content that has been created, generated, composed, or produced for delivery. The external content 38 may include any other external content produced elsewhere or stored in the subsystem 20.

The network 40 provides interconnections to the delivery subsystem 20, the receiving subsystem 50, the Web site 60, and other nodes or subsystems. The network 40 may correspond to any topologies and protocols. Some examples are Internet, Local Area Network, Wide Area Network, broadcast network, etc. The transmission medium may be any suitable communication medium such as air, cable, electromagnetic, optic, etc.

The receiving subsystem 50 receives the packaged content delivered by the delivery subsystem 20. The receiving subsystem 50 may be similar to the delivery subsystem 50 and may include a server, a computer system, and a platform. An embodiment of the receiving subsystem 50 is illustrated in Figure 1B. The receiving subsystem 50 includes a packaged content receiver 55 that performs the receiving function. The receiving subsystem 50 provides interface to a user to use the received packaged content efficiently and intelligently.

The Web site 60 provides accesses to information, content, data, and files for downloading and capturing. The Web site 60 may be a source to provide the content to the delivery subsystem 20.

Figure 1B is a diagram illustrating a computer system 100 according to one embodiment of the present invention. The computer system 100 may be used as part of the delivery subsystem 20 or the receiver subsystem 50. The computer system 100 includes a processor 105, a host bus 110, a host bridge chipset 120, a system memory 130, a peripheral component interconnect (PCI) bus 155, K PCI slots $160_1$ to $160_K$, a PCI-to-industry standard adapter (ISA) bridge 170, mass storage devices 172, Input/Output (I/O) ports 174, an ISA bus 180, and ISA slots $185_1$ to $185_M$.

The processor 105 represents a central processing unit of any type of architecture, such as complex instruction set computers (CISC), reduced instruction set computers (RISC), very long instruction word (VLIW), or hybrid architecture. The processor 105 interfaces with the host bridge chipset 120 via the host bus 110. The host bus 110 may support single processor or multiple processors.

The host bridge chipset 120 includes a number of interface circuits to allow the host processor 105 access to the system memory 130 and the PCI bus 155. The system memory 130 represents one or more mechanisms for storing information. For example, the system memory 130 may include non-volatile or volatile memories. Examples of these memories include flash memory, read only memory (ROM), or random access memory (RAM). The system memory 130 may contain the packaged content delivery module 25 or the packaged content receiver 55 (Figure 1A), and other programs and data 138. Of course, the system memory 130 preferably contains additional software (not shown), which is not necessary to understanding the invention.

The PCI slots $160_1$ to $160_K$ provide interfaces to PCI devices. Examples of PCI devices include the network interface and the media interface. The network interface connects to communication networks such as the Internet. The Internet provides access to on-line service providers, Web browsers, and other network channels. The media interface provides access to audio and video devices.

The PCI-to-ISA bridge 170 provides access to the ISA bus 180, mass storage devices 172, and I/O ports 174. The mass storage devices 172 include CD-ROM, floppy diskettes, and hard drives. The ISA bus 180 has a number of ISA slots $185_1$ to $185_M$ to interface to ISA devices. Examples of ISA devices include data entry devices (e.g., keyboard, mouse), printers, etc.

When implemented in software, the elements of the present invention are essentially the code segments to perform the necessary tasks. The program or code segments can be stored in a processor readable medium or transmitted by a computer data signal embodied in a carrier wave, or a signal modulated by a carrier, over a transmission medium. The "processor readable medium" may include any medium that can store or transfer information such as the mass storage devices 172. Examples of the processor readable medium include an electronic circuit, a semiconductor memory device, a ROM, a flash memory, an erasable ROM (EROM), a floppy diskette, a compact disk CD-ROM, an optical disk, a hard disk, a fiber optic medium, a radio frequency (RF) link, etc. The computer data signal may include any signal that can propagate over a transmission medium such as electronic network channels, optical fibers, air, electromagnetic, RF links, etc. The code segments may be downloaded via computer networks such as the Internet, Intranet, etc.

The packaged content delivery module 25 or the packaged content receiver 55 may have several modules coupled to one another. A software module is coupled to another module to receive variables, parameters, arguments, pointers, etc. and/or to generate or pass results, updated variables, pointers, etc. A software module may also be a software driver or interface to interact with the operating system running on the platform. A software module may also be a hardware driver to configure, set up, initialize, send and receive data to and from a hardware device.

Figure 2 is a diagram illustrating the packaged content delivery module 25 shown in Figure 1A according to one embodiment of the invention. The packaged content delivery module 25 includes a package generator 210, a cache manager 230, and a native file system 250. The packaged content delivery module 25 may be implemented by hardware, software, or a combination of both hardware and software.

The package generator 210 collects the content from the content sources 30 and generates the packaged content 27 for delivery to the receiving subsystem 50 (Figure 1A). The package generator 210 includes a package composer 212, a packaged content tagger 214, and a packaged content serializer 216. The package composer 212 composes a package from the collected content. The packaged content tagger 214 tags the package and its package elements with meta-data, which describes or characterizes the corresponding package elements and the package. The packaged content serializer 216 produced the packaged content ready for delivery with appropriate markup language.

The cache manager 230 manages cache memories of the platform on which the package generator 210 operates. The cache manager 230 provides storage mechanism for the content and the meta-data used by the package generator 210. The file system 250 provides an environment to organize data, content, files into an hierarchical file organization. The file system 250 provides directories, subdirectories and other file facilities to facilitate the file management.

Figure 3 is a diagram illustrating the package composer 212 shown in Figure 2 according to one embodiment of the invention. The package composer 212 includes a collector 310, and an organizer 330.

The collector 310 collects or gathers the content from the content sources 30 into a post-production site to create a package 320. The content sources 30 may include the Web content 32, the re-purposed Web content 34, the produced content 36, and other external content sources 38. The content may include HyperText Markup Language (HTML) pages, MP3 files, QuickTime movies, video clips, or any other suitable content. Any content source or provider may act as a content originator or a content aggregator depending on the exact nature of the business models and alliances with other content sources.

The organizer 330 organizes the collected content into N package elements $335_1$ to $335_N$ for the package 320. Each package element $335_k$ corresponds to a content type. For example, the package element $335_1$ may correspond to an MP3 file, the package element $335_2$ may correspond to an HTML page, the package element $335_3$ may correspond to another HTML page, etc. The organizer 330 provides a unique identifier to each of the package elements $335_1$ to $335_N$. The organizer 330 also interacts with the file system 250 (Figure 2) to store the package elements into a set of files or directories/ subdirectories that contain audio, video, data, script, executables, and other content types.

Figure 4 is a diagram illustrating the packaged content tagger 214 shown in Figure 2 according to one embodiment of the invention. The packaged content tagger 214 includes an element tagger 410, a package element grouper 450, a package tagger 470, a meta-data 430, and a package marker 490.

The element tagger 410 and the package tagger 470 interface with the meta-data 430 to obtain tags that characterize the package elements $335_1$ to $335_N$ and the package 320. The element tagger 410 assigns a tag from the meta-data 430 to each of the package elements.

The meta-data 430 includes tags or descriptors for the package elements and the package 320. The tags for the package elements include an element name tag 432, an element location tag 434, a content originator tag 436, a content expiration tag 438, a content presentation tag 440, a content filtering tag 442, a platform tag 444, and a personalization tag 446. The tags for the package 320 include a package identity tag 448. The element name tag 432 is the name of the package element. The element location tag 434 indicates the location of the package element within the package according to some pre-defined location protocol. For example, the element location tag 434 may be an absolute address, a relative address, or a pointer to the corresponding package element. The content originator tag 436 provides information about the content provider that provides the content corresponding to the package element. The information may include the identifier, URL, and content type. The content expiration tag 438 indicates the content lifetime within the network. The content presentation tag 440 is used by the presentation engine in the receiver subsystem for rendering the end-user experience. The content filtering tag 442 indicates how the content should be categorized for filtering. The content filtering tag 442 may be used by heuristics that range from simple matching to correlation matching. The platform tag 444 is used to tailor the package element to specific platform capabilities. The content personalization tag 446 is used to personalize the content filtering and presentation to personal preferences for users on the receiver subsystem. The package identity tag 448 indicates the package identifier and/or the package element identifiers.

The package element grouper 450 groups the tagged package elements into logical groups based on some pre-defined criteria. For example, package elements having the same tags may be grouped together. The package element grouper 450 generates grouped tagged package elements 455.

The package tagger 470 tags the package 320 with the corresponding package tags. The package marker 490 marks the tagged package for inclusion in a playlist 495 that becomes part of the packaged content to be delivered.

Figure 5 is a diagram illustrating the packaged content serializer 216 shown in Figure 2 according to one embodiment of the invention. The packaged content serializer 216 includes a representer 510, a schema 530, and a transformer 550.

The representer 510 represents the element tags and the package tag for the package 320 using the schema 530. The schema 530 provides a framework for the representation. Typically, the schema 530 follows some universal standard or protocol so that the delivery packaged content becomes independent of the network and the transmission medium mechanisms and policies. In one embodiment, the schema 530 is based on the extensible markup language (XML). The representer 510 generates markup language elements $515_1$ to $515_N$ and markup language package 518 corresponding to the element tags $415_1$ to $415_N$ and the package tag 475, respectively.

The transformer 550 combines the markup language elements $515_1$ to $515_N$, the markup language package 518 and the package 320 or the playlist 495 with all the package elements $335_1$ to $335_N$ and transform into a serialized package markup language packaged content 555. The serialized package markup language packaged content 555 becomes the delivery packaged content 27 (Figure 1) to be delivered to the receiving subsystem 50. The delivery packaged content 27 is also compatible with the file system 250 (Figure 2) which is compatible with the file system of the receiver subsystem 50.

Figure 6 is a diagram illustrating the packaged content receiver 55 shown in Figure 1 according to one embodiment of the invention. The packaged content receiver 55 includes a receiver interface 610,a receiver meta-data 618, user's settings 615, a receiver organizer 630 a cache manager 650, and a receiver file system 660.

The receiver interface 610 receives the delivery packaged content 27 and parses the delivery packaged content 27 according to some pre-defined preferences such as receiver user's settings 615. The receiver interface 610 may also tags the delivery packaged content 27 with the receiver meta-data 618. The receiver interface 610 then generates a pre-processed package 620 having content modified according to the receiver user's settings 615. For example, the receiver interface 610 may retain the package elements that have the filtering tags matching the user's filtering criteria set at the receiver subsystem 50. As another example, the receiver interface 610 may discard the package elements that have the expiration tags (e.g., obsolete news articles) exceeding the limit set by the receiver user.

The receiver organizer 630 organizes the pre-processed package 620 according to the receiver file system 660. The cache manager 650 manages cache memories of the platform on which the receiver organizer 630 operates. The cache manager 650 provides storage mechanism for the organized package. The receiver organizer 630 also uses the tagged receiver meta-data when organizing the packaged content. By sorting out the packaged content according to well-defined criteria and settings, the packaged content receiver 55 allows the receiver user to access the packaged content in an intelligent manner, personalized to user's preferences and criteria.

While this invention has been described with reference to illustrative embodiments, this description is not intended to be construed in a limiting sense. Various modifications of the illustrative embodiments, as well as other embodiments of the invention, which are apparent to persons skilled in the art to which the invention pertains are deemed to lie within the spirit and scope of the invention.

## CLAIMS

What is claimed is:

1. A method comprising:

composing a packaged content for delivery over a network to a subsystem, the packaged content having a package containing content, the package having package elements;

tagging the package and the package elements using meta-data; and

serializing the meta-data into a file system compatible with the subsystem using a markup language.

2. The method of claim 1 wherein composing comprises:

collecting the content for the package elements from a plurality of content sources; and

organizing the package elements into a set of files.

3. The method of claim 2 wherein the content source is one of web content, re-purposed web content, produced content, and external content.

4. The method of claim 3 wherein the content source is one of a video, a data, a script, and an executable content.

5. The method of claim 2 wherein organizing comprises:

providing a unique identifier to each of the package elements; and storing the package elements in the file system corresponding to the set of files.

6. The method of claim 1 wherein tagging comprises:

assigning an element tag to each of the package elements;

grouping the package elements according to the corresponding element tags;

assigning a package tag to the package based on the grouped package elements, the package tag and the element tags forming the meta-data; and

marking the package for inclusion into a playlist.

7. The method of claim 6 wherein the element tag is one of an element location tag, an element name tag, a content originator tag, a content presentation tag, a content filtering tag, a platform tag, and a personalization tag.

8. The method of claim 1 wherein serializing comprises:

representing the meta-data using a schema based on the markup language; and

transforming the represented meta-data and the package into the packaged content corresponding to the file system.

9. The method of claim 8 wherein representing the meta-data comprises:

representing the meta-data using an extensible markup language (XML) schema.

10. A method comprising:

processing a packaged content delivered over a network from a delivery subsystem, the packaged content having a package encapsulated with associated meta-data using a markup language, the package having package elements corresponding to a content; and

organizing the processed packaged content according to a receiver file system.

11. The method of claim 10 wherein processing comprises:

parsing the packaged content to extract the package elements; and

sorting the package elements according to the meta-data.

12. The method of claim 11 wherein sorting comprises:

matching the meta-data with user's settings.

13. A computer program product comprising:

a machine useable medium having computer program code embedded therein, the computer program product having:

computer readable program code to compose a packaged content for delivery over a network to a subsystem, the packaged content having a package containing content, the package having package elements;

computer readable program code to tag the package and the package elements using meta-data; and

computer readable program code to serialize the meta-data into a file system compatible with the subsystem using a markup language.

14. The computer program product of claim 13 wherein the computer readable program code to compose comprises:

computer readable program code to collect the content for the package elements from a plurality of content sources; and

computer readable program code to organize the package elements into a set of files.

15. The computer program product of claim 14 wherein the content source is one of web content, re-purposed web content, produced content, and external content.

16. The computer program product of claim 15 wherein the content source is one of a video, a data, a script, and an executable content.

17. The computer program product of claim 14 wherein the computer readable program code to organize comprises:

computer readable program code to provide a unique identifier to each of the package elements; and

computer readable program code to store the package elements in the file system corresponding to the set of files.

18. The computer program product of claim 13 wherein the computer readable program code to tag comprises:

computer readable program code to assign an element tag to each of the package elements;

computer readable program code to group the package elements according to the corresponding element tags;

computer readable program code to assign a package tag to the package based on the grouped package elements, the package tag and the element tags forming the meta-data; and

computer readable program code to mark the package for inclusion into a playlist.

19. The computer program product of claim 18 wherein the element tag is one of an element location tag, an element name tag, a content originator tag, a content presentation tag, a content filtering tag, a platform tag, and a personalization tag.

20. The computer program product of claim 13 wherein the computer readable program code to serialize comprises:

computer readable program code to represent the meta-data using a schema based on the markup language; and

computer readable program code to transform the represented meta-data and the package into the packaged content corresponding to the file system.

21. The computer program product of claim 20 wherein the computer readable program code to represent the meta-data comprises:

computer readable program code to represent the meta-data using an extensible markup language (XML) schema.

22. A computer program product comprising:

a machine useable medium having computer program code embedded therein, the computer program product having:

computer readable program code to process a packaged content delivered over a network from a delivery subsystem, the packaged content having a package encapsulated with associated meta-data using a markup language, the package having package elements corresponding to a content; and

computer readable program code to organize the processed packaged content according to a receiver file system.

23. The computer program product of claim 22 wherein the computer readable program code to process comprises:

computer readable program code to parse the packaged content to extract the package elements; and

computer readable program code to sort the package elements according to the meta-data.

24. The computer program product of claim 23 wherein the computer readable program code to sort comprises:

computer readable program code to match the meta-data with user's settings.

25. A system comprising:

a file system;

a cache manager coupled to the file system to manage a cache memory; and

a packaged content delivery module coupled to the cache manager to deliver a packaged content to a subsystem over a network, the packaged content delivery module comprising:

a composer to compose the packaged content, the packaged content having a package containing content, the packaging having package elements,

a tagger coupled to the composer to tag the package and the package elements using meta-data, and

a serializer coupled to the tagger to serialize the meta-data into the file system compatible with the subsystem using a markup language.

26. The system of claim 25 wherein the composer comprises:

a collector to collect the content for the package elements from a plurality of content sources; and

an organizer coupled to the collector to organize the package elements into a set of files.

27. The system of claim 26 wherein the content source is one of web content, re-purposed web content, produced content, and external content.

28. The system of claim 27 wherein the content source is one of a video, a data, a script, and an executable content.

29. The system of claim 26 wherein the organizer provides a unique identifier to each of the package elements; and stores the package elements in the file system corresponding to the set of files.

30. The system of claim 25 wherein the tagger comprises:

an element tagger to assign an element tag to each of the package elements;

an element grouper coupled to the element tagger to group the package elements according to the corresponding element tags;

a package tagger to assign a package tag to the package based on the grouped package elements, the package tag and the element tags forming the meta-data; and

a package marker coupled to the package tagger to mark the package for inclusion into a playlist.

31. The system of claim 30 wherein the element tag is one of an element location tag, an element name tag, a content originator tag, a content presentation tag, a content filtering tag, a platform tag, and a personalization tag.

32. The system of claim 25 wherein the serializer comprises:

a representer to represent the meta-data using a schema based on the markup language; and

a transformer coupled to the representer to transform the represented meta-data and the package into the packaged content corresponding to the file system.

33. The system of claim 32 wherein the representer comprises:

a extensible markup language (XML) representer to represent the meta-data using an XML schema.

34. A system comprising:

a storage to store user settings;

a receiver interface coupled to the storage to process a packaged content according to the user settings, the packaged content being delivered over a network from a delivery subsystem, the packaged content having a package encapsulated with associated meta-data using a markup language, the package having package elements corresponding to a content; and

a receiver organizer coupled to the receiver interface to organize the processed packaged content according to a receiver file system.

35. The system of claim 34 wherein the receiver interface comprises:

a parser to parse the packaged content, the parser extracting the package elements; and

a sorter coupled to the parser to sort the package elements according to the meta-data.

36. The system of claim 35 wherein the sorter comprises:

a matcher to match the meta-data with the user settings.

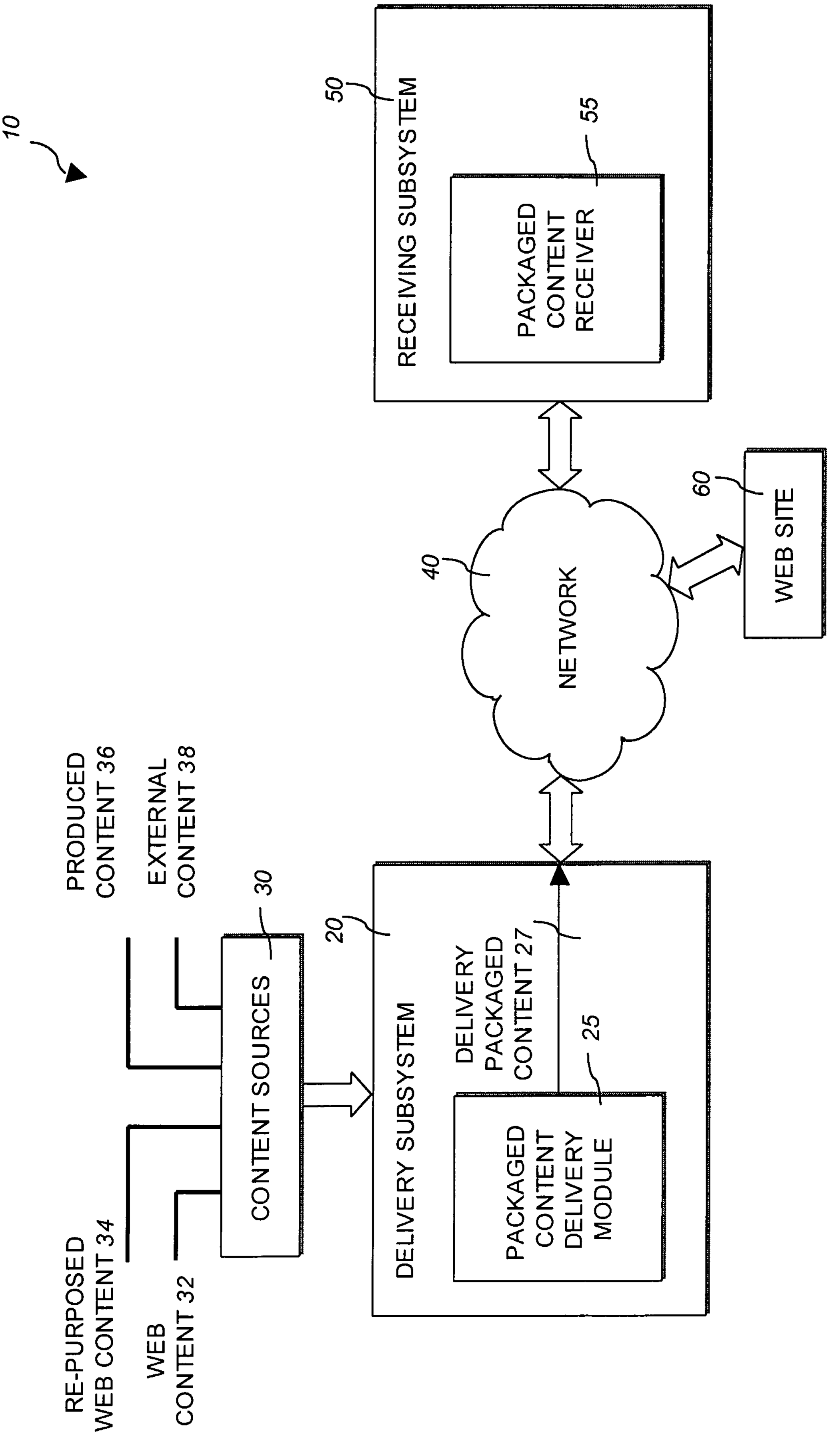

10
RE-PURPOSED WEB CONTENT 34
WEB CONTENT 32
PRODUCED CONTENT 36
EXTERNAL CONTENT 38
CONTENT SOURCES
30
DELIVERY SUBSYSTEM
20
PACKAGED CONTENT DELIVERY MODULE
25
DELIVERY PACKAGED CONTENT 27
NETWORK
40
WEB SITE
60
RECEIVING SUBSYSTEM
50
PACKAGED CONTENT RECEIVER
55

FIG.1A

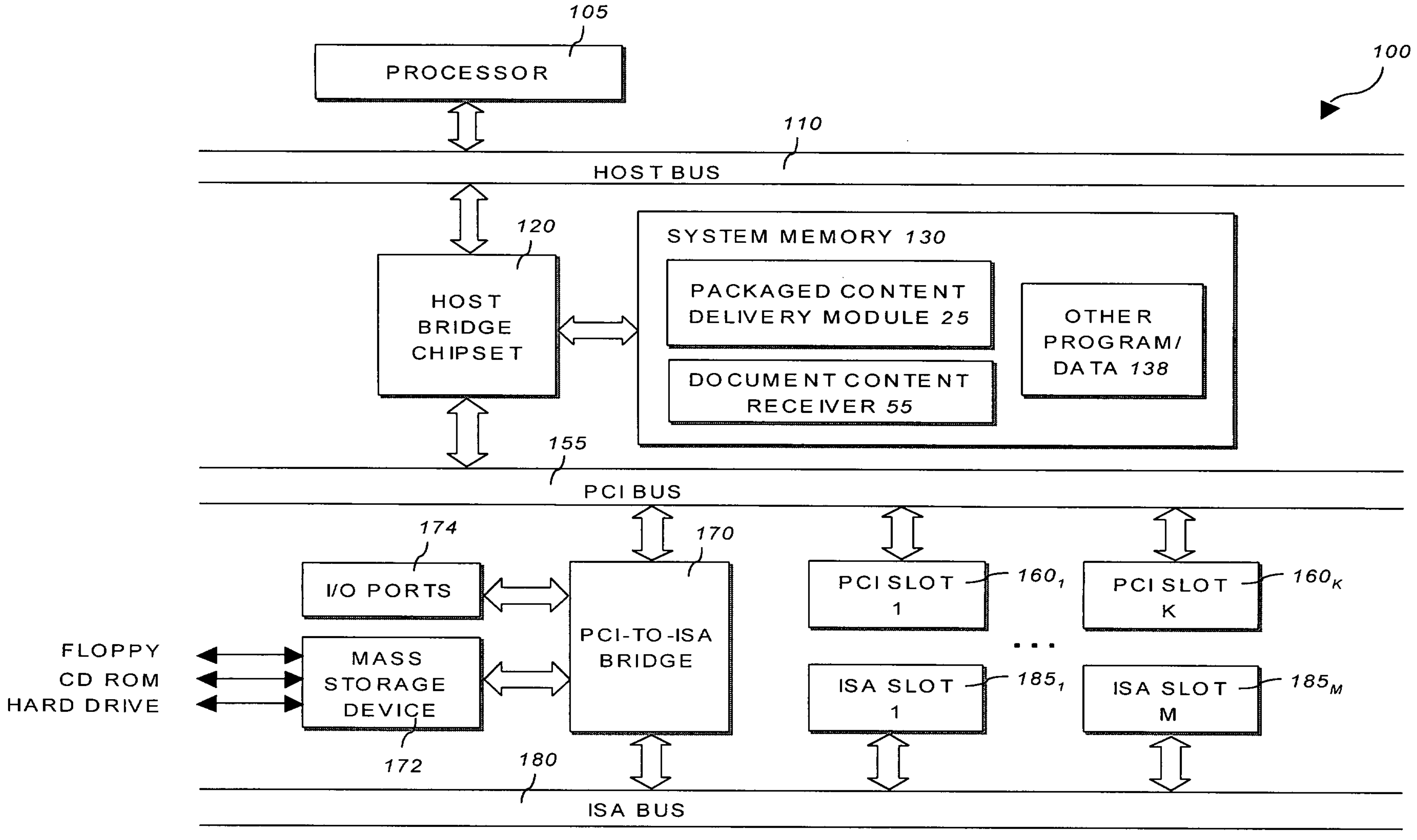

100
105
PROCESSOR
110
HOST BUS
120
HOST BRIDGE CHIPSET
SYSTEM MEMORY 130
PACKAGED CONTENT DELIVERY MODULE 25
DOCUMENT CONTENT RECEIVER 55
OTHER PROGRAM/ DATA 138
155
PCI BUS
174
I/O PORTS
170
PCI-TO-ISA BRIDGE
PCI SLOT 1
$160_1$
PCI SLOT K
$160_K$
. . .
FLOPPY
CD ROM
HARD DRIVE
MASS STORAGE DEVICE
172
ISA SLOT 1
$185_1$
ISA SLOT M
$185_M$
180
ISA BUS

FIG.1B

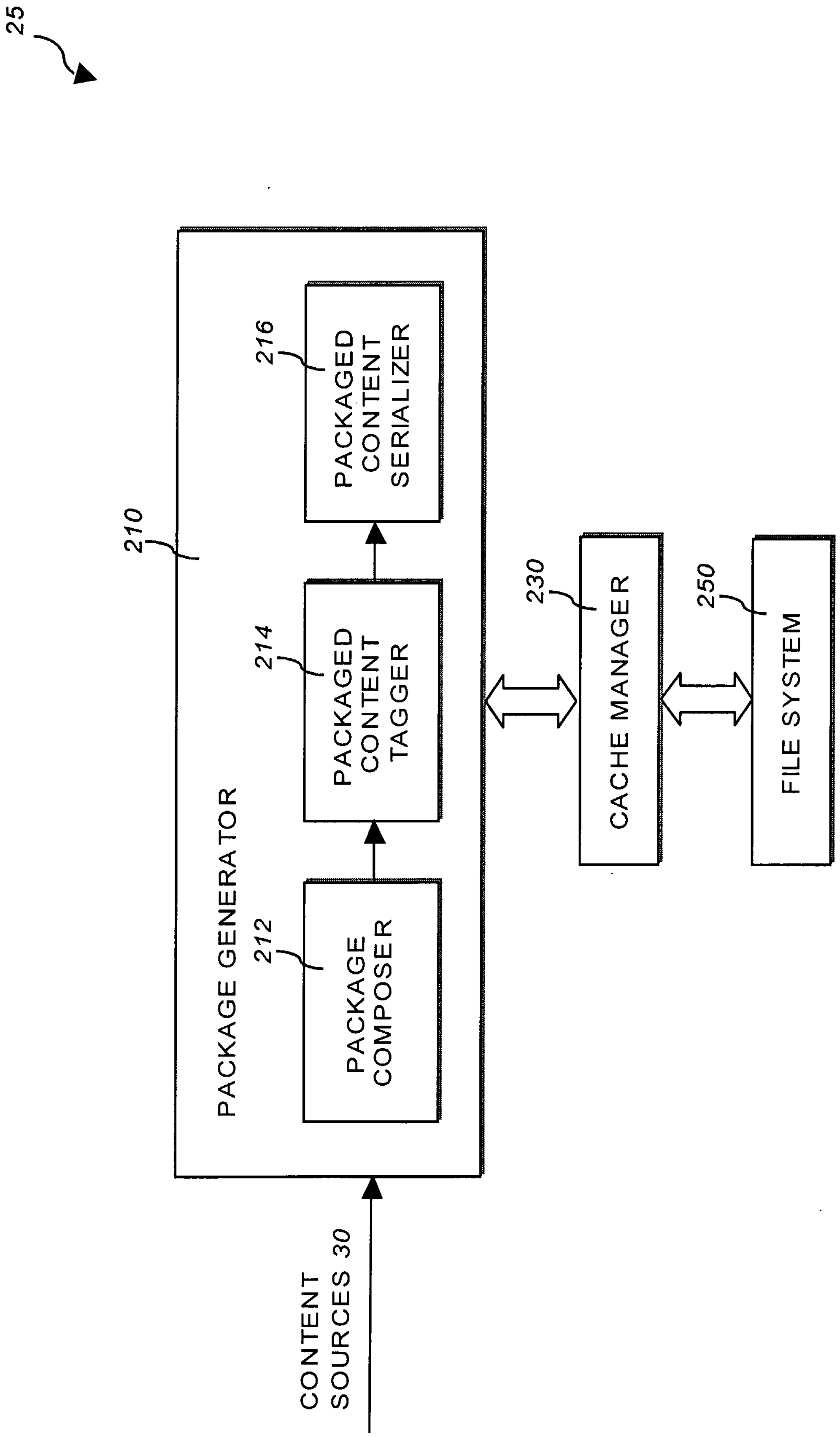

25
210
PACKAGE GENERATOR
212
PACKAGE COMPOSER
214
PACKAGED CONTENT TAGGER
216
PACKAGED CONTENT SERIALIZER
230
CACHE MANAGER
250
FILE SYSTEM
CONTENT SOURCES 30

FIG.2

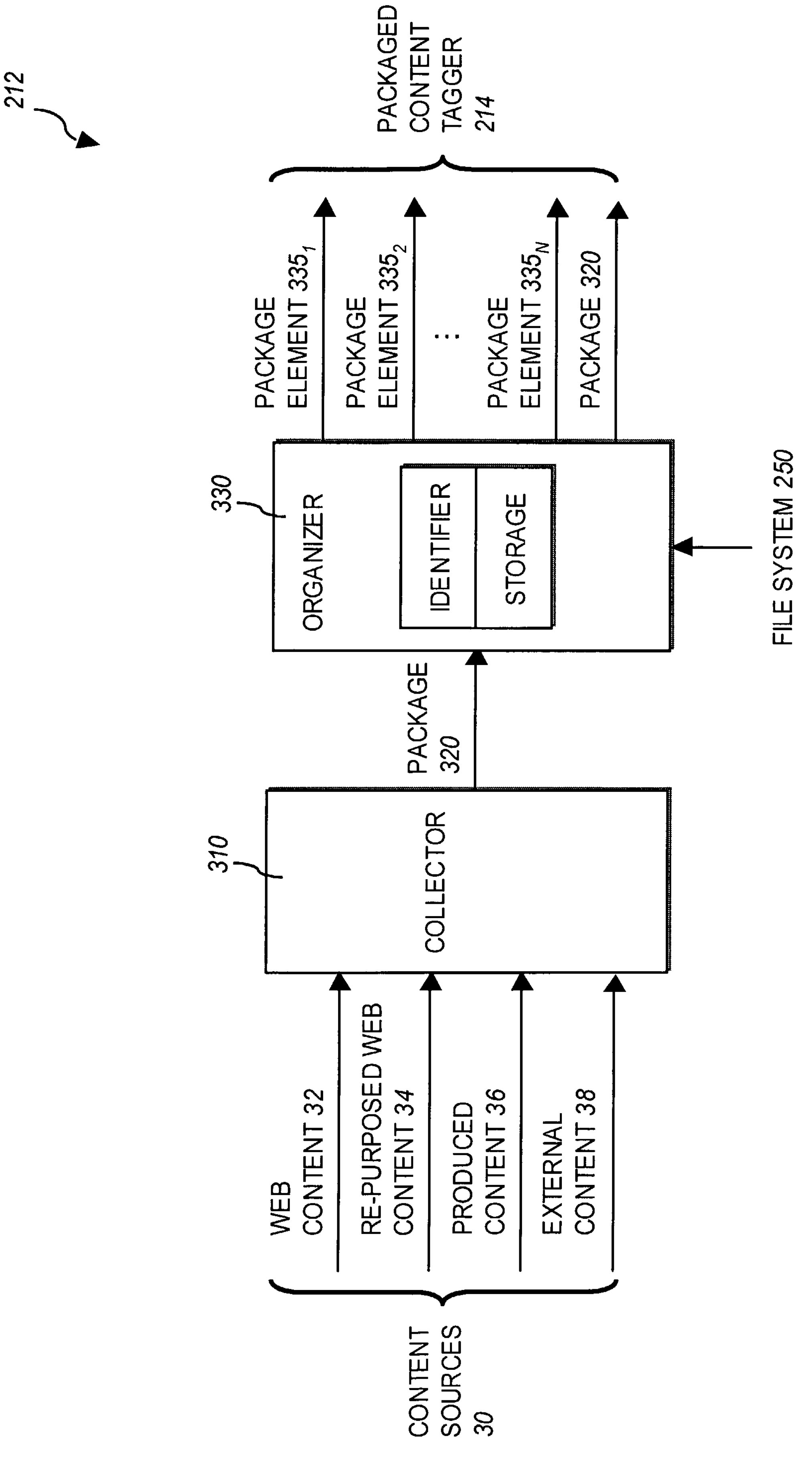

212
PACKAGED CONTENT TAGGER 214
PACKAGE ELEMENT $335_1$
PACKAGE ELEMENT $335_2$
⋮
PACKAGE ELEMENT $335_N$
PACKAGE 320
330
ORGANIZER
IDENTIFIER
STORAGE
FILE SYSTEM 250
PACKAGE 320
310
COLLECTOR
WEB CONTENT 32
RE-PURPOSED WEB CONTENT 34
PRODUCED CONTENT 36
EXTERNAL CONTENT 38
CONTENT SOURCES 30

FIG.3

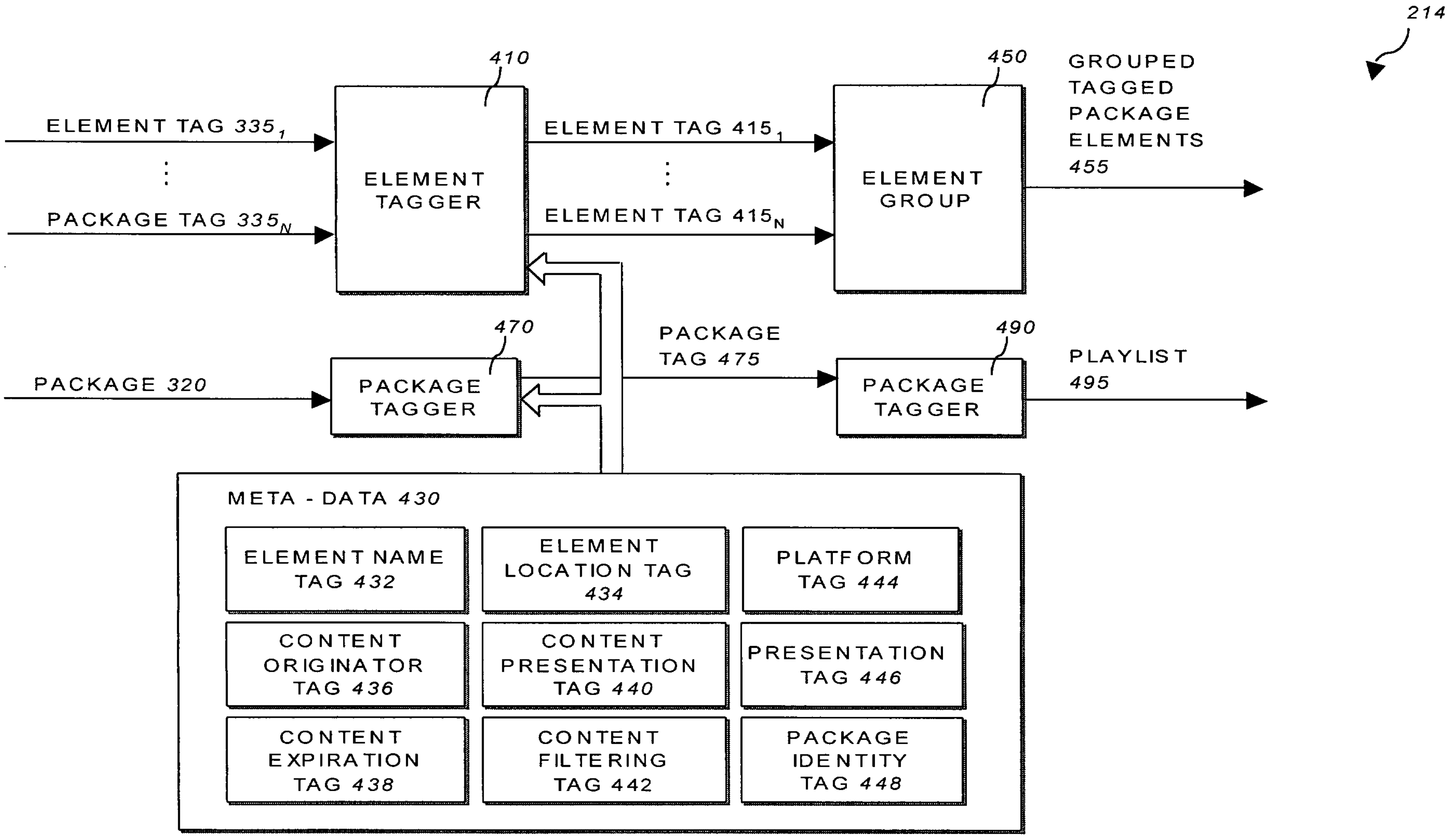

214
ELEMENT TAG $335_1$
⋮
PACKAGE TAG $335_N$
410
ELEMENT TAGGER
ELEMENT TAG $415_1$
⋮
ELEMENT TAG $415_N$
450
ELEMENT GROUP
GROUPED TAGGED PACKAGE ELEMENTS 455
PACKAGE 320
470
PACKAGE TAGGER
PACKAGE TAG 475
490
PACKAGE TAGGER
PLAYLIST 495
META - DATA 430
ELEMENT NAME TAG 432
ELEMENT LOCATION TAG 434
PLATFORM TAG 444
CONTENT ORIGINATOR TAG 436
CONTENT PRESENTATION TAG 440
PRESENTATION TAG 446
CONTENT EXPIRATION TAG 438
CONTENT FILTERING TAG 442
PACKAGE IDENTITY TAG 448

FIG. 4

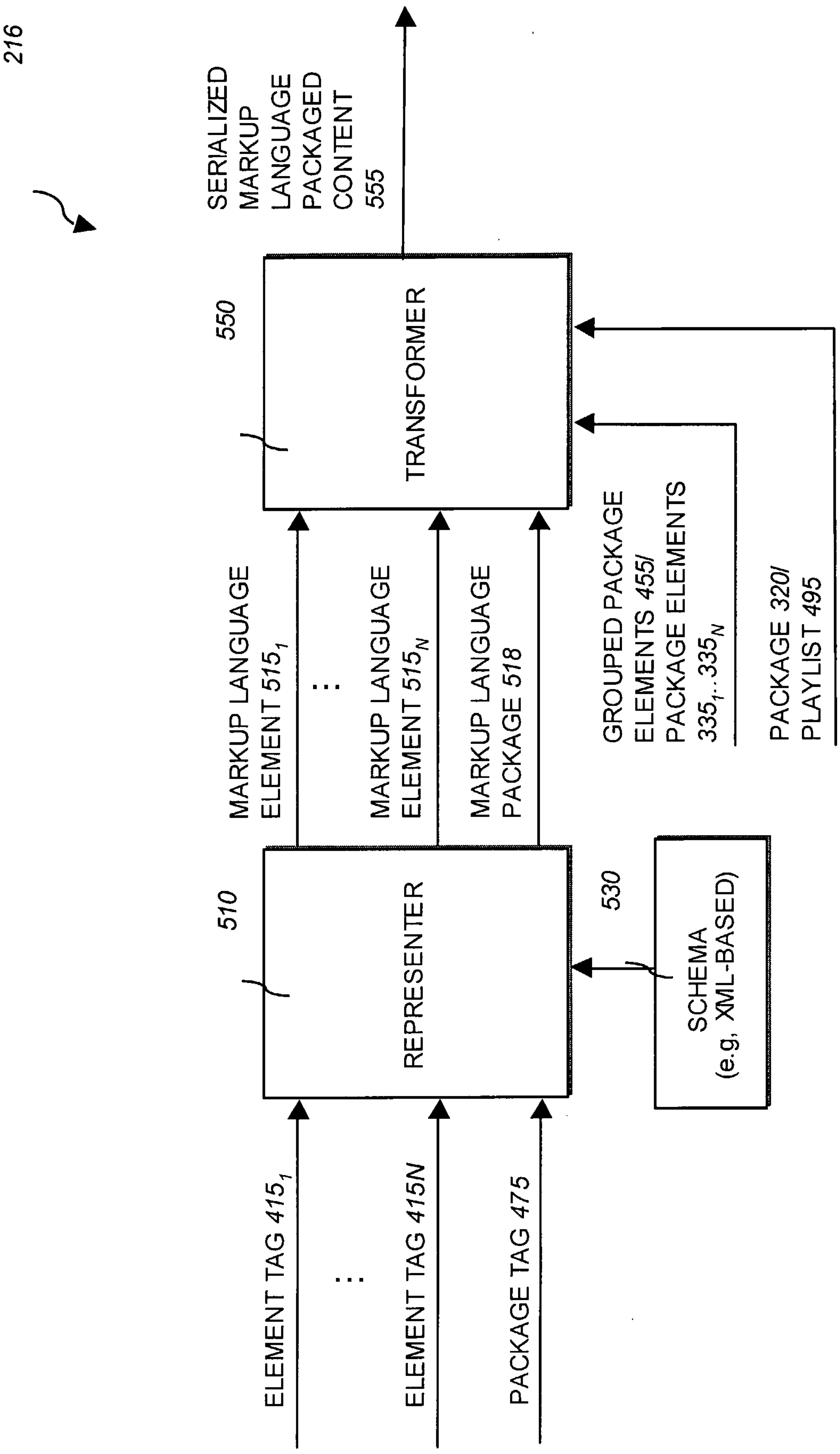

216
ELEMENT TAG $415_1$
...
ELEMENT TAG 415N
PACKAGE TAG 475
510
REPRESENTER
530
SCHEMA (e.g, XML-BASED)
MARKUP LANGUAGE ELEMENT $515_1$
...
MARKUP LANGUAGE ELEMENT $515_N$
MARKUP LANGUAGE PACKAGE 518
GROUPED PACKAGE ELEMENTS 455/ PACKAGE ELEMENTS $335_1..335_N$
PACKAGE 320/ PLAYLIST 495
550
TRANSFORMER
SERIALIZED MARKUP LANGUAGE PACKAGED CONTENT 555

FIG.5

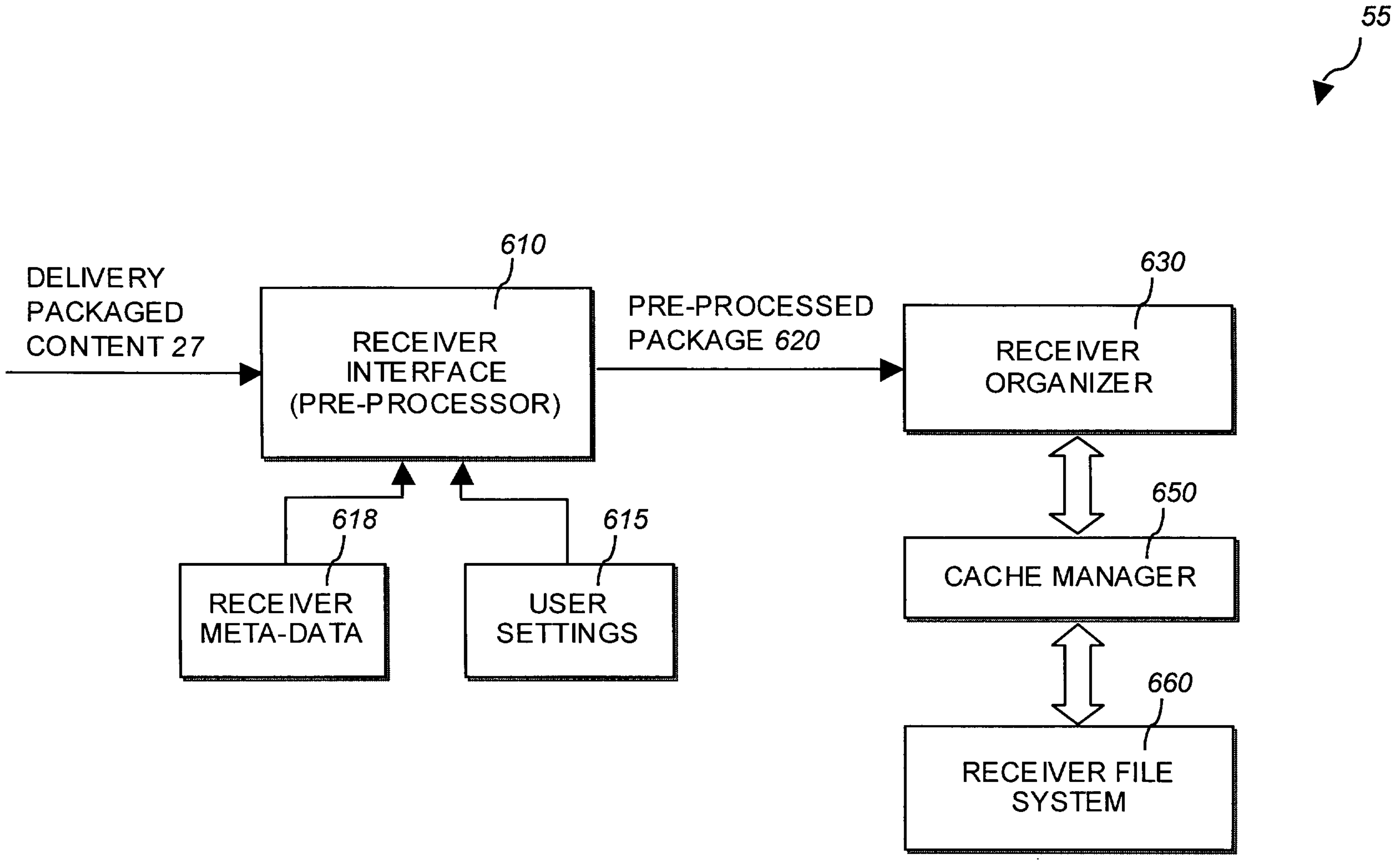

55
DELIVERY
PACKAGED
CONTENT 27
610
RECEIVER
INTERFACE
(PRE-PROCESSOR)
PRE-PROCESSED
PACKAGE 620
630
RECEIVER
ORGANIZER
618
RECEIVER
META-DATA
615
USER
SETTINGS
650
CACHE MANAGER
660
RECEIVER FILE
SYSTEM

FIG.6