

(12) DEMANDE DE BREVET CANADIEN  
CANADIAN PATENT APPLICATION

(13) **A1**

(86) Date de dépôt PCT/PCT Filing Date: 2002/04/18  
(87) Date publication PCT/PCT Publication Date: 2002/10/31  
(85) Entrée phase nationale/National Entry: 2003/10/14  
(86) N° demande PCT/PCT Application No.: US 2002/012307  
(87) N° publication PCT/PCT Publication No.: 2002/087231  
(30) Priorités/Priorities: 2001/04/20 (09/838,782) US;  
2002/04/04 (10/115,693) US

(51) Cl.Int.<sup>7</sup>/Int.Cl.<sup>7</sup> G11B 27/034, G11B 27/34

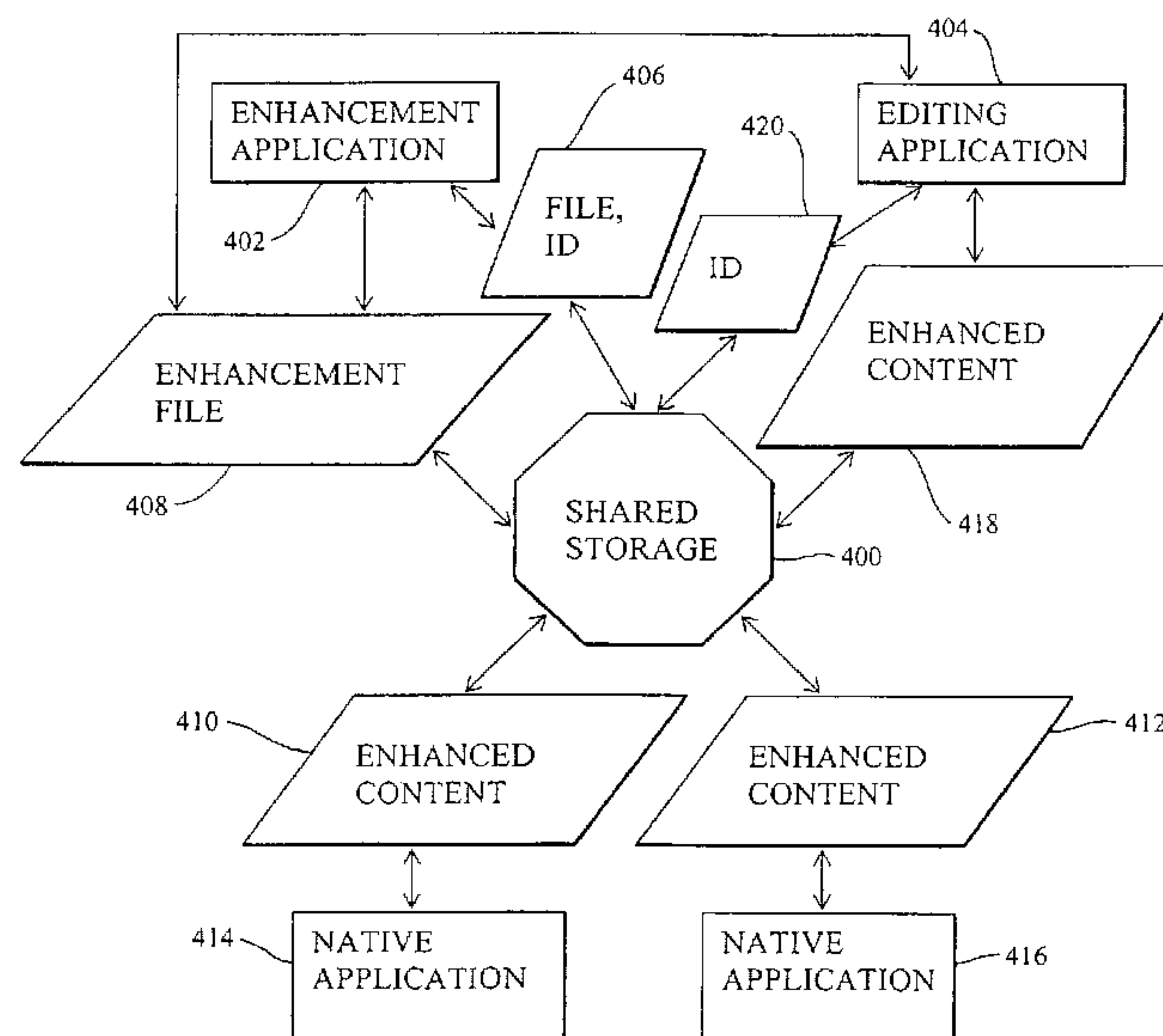
(71) Demandeur/Applicant:  
AVID TECHNOLOGY, INC., US

(72) Inventeurs/Inventors:  
PHILLIPS, MICHAEL, US;  
COOPER, BRIAN C., US;  
FAY, LARISA, US

(74) Agent: SMART & BIGGAR

(54) Titre : EDITION DE SUPPORTS TEMPORELS A CONTENU AMELIORE

(54) Title: EDITING TIME-BASED MEDIA WITH ENHANCED CONTENT



(57) **Abrégé/Abstract:**

Creation of a program with interactive content and time-based media would be improved by having several people working simultaneously on both the interactive content and the time-based media. The range of types of data that can be associated with the time-based media also can include a variety of types of content that enhance the time-based media, including data that is opaque to the editing system. To support editing of such programs, enhanced content may be edited into a program as either a clip or a locator, according to a user's selection. The components that represent enhanced content also may be edited in the same way as components that represent time-based media. Enhanced content may be represented by enhancement files that include data defining enhancements. An enhancement includes an identifier of the enhancement within the enhancement file, and a reference to a file defining enhanced content associated with the enhancement. Such enhancement files may be imported into a bin of an editing system to create components that describe the enhanced content referenced by the enhancements. The enhanced content may be modified using native applications for the enhanced content during editing of the program. The components describing the enhanced content in the editing system can be updated using identifiers of the enhanced content that are stored by the components. Enhanced content also may be automatically synchronized with time-based media.

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

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
31 October 2002 (31.10.2002)

PCT

(10) International Publication Number  
**WO 02/087231 A3**

(51) International Patent Classification<sup>7</sup>: **G11B 27/034**,  
27/34

Balcom Drive, Foxboro, MA 02035 (US). **FAY, Larisa**; 2  
Pole Hill Drive, Andover, MA 01810 (US).

(21) International Application Number: PCT/US02/12307

(74) Agent: **GORDON, Peter, J.**; Avid Technology, Inc., One  
Park West, Tewksbury, MA 01876 (US).

(22) International Filing Date: 18 April 2002 (18.04.2002)

(81) Designated States (*national*): CA, DE, GB, JP.

(25) Filing Language: English

(84) Designated States (*regional*): European patent (AT, BE,  
CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC,  
NL, PT, SE, TR).

(26) Publication Language: English

(30) Priority Data:

09/838,782 20 April 2001 (20.04.2001) US  
10/115,693 4 April 2002 (04.04.2002) US

**Published:**  
— with international search report

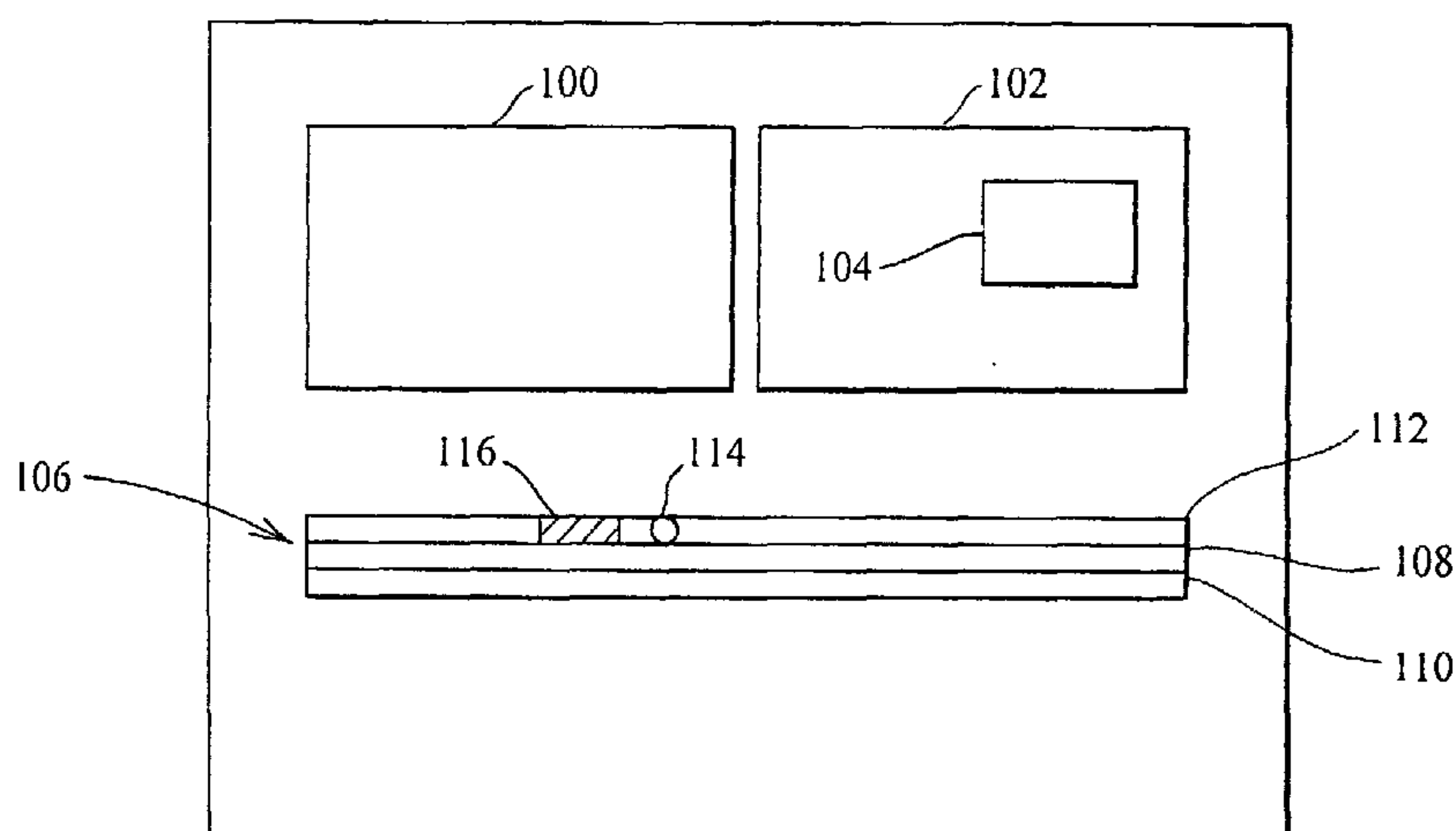
(71) Applicant: **AVID TECHNOLOGY, INC.** [—/US]; Avid  
Technology Park, One Park West, Tewksbury, MA 01876  
(US).

(88) Date of publication of the international search report:  
6 February 2003

(72) Inventors: **PHILLIPS, Michael**; 94 Hillside Avenue,  
Melrose, MA 02176-3218 (US). **COOPER, Brian, C.**; 8

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

(54) Title: EDITING TIME-BASED MEDIA WITH ENHANCED CONTENT



(57) **Abstract:** Creation of a program with interactive content and time-based media would be improved by having several people working simultaneously on both the interactive content and the time-based media. The range of types of data that can be associated with the time-based media also can include a variety of types of content that enhance the time-based media, including data that is opaque to the editing system. To support editing of such programs, enhanced content may be edited into a program as either a clip or a locator, according to a user's selection. The components that represent enhanced content also may be edited in the same way as components that represent time-based media. Enhanced content may be represented by enhancement files that include data defining enhancements. An enhancement includes an identifier of the enhancement within the enhancement file, and a reference to a file defining enhanced content associated with the enhancement. Such enhancement files may be imported into a bin of an editing system to create components that describe the enhanced content referenced by the enhancements. The enhanced content may be modified using native applications for the enhanced content during editing of the program. The components describing the enhanced content in the editing system can be updated using identifiers of the enhanced content that are stored by the components. Enhanced content also may be automatically synchronized with time-based media.



**WO 02/087231 A3**

## EDITING TIME-BASED MEDIA WITH ENHANCED CONTENT

## CROSS REFERENCE TO RELATED APPLICATIONS

This application claims the benefit under 35 U.S.C. §120, and is a continuing  
5 application, of U.S. Patent Application 09/838,782, filed April 20, 2001, now pending.

## BACKGROUND

Programs that combine video and audio with interactive content generally are  
created using one of two approaches. The first approach involves creating audiovisual  
10 content as an element of the interactive program, in which interactive content refers to the  
audiovisual content. The other approach involves creating an audiovisual program, and  
then associating interactive content with different points in time in the audiovisual  
program. In both such approaches, the audiovisual content is created first and then is  
provided to editors of interactive content who embellish the audiovisual content with  
15 interactive content to produce the final interactive program. The interactive content  
typically is limited to documents in a standard format to be displayed or machine  
instructions to be executed in conjunction with playback of the audiovisual content.

## SUMMARY

20 Creation of a program with interactive content and time-based media would be  
improved by having several people working simultaneously on both the interactive  
content and the time-based media. The range of types of data that can be associated with  
the time-based media also can include a variety of types of content that enhance the time-  
based media, including data that is opaque to the editing system. To support editing of  
25 such programs, enhanced content may be edited into a program as either a clip or a  
locator, according to a user's selection. The components that represent enhanced content  
also may be edited in the same way as components that represent time-based media.  
Enhanced content may be represented by enhancement files that include data defining  
enhancements. An enhancement includes an identifier of the enhancement within the  
30 enhancement file, and a reference to a file defining enhanced content associated with the  
enhancement. Such enhancement files may be imported into a bin of an editing system to

create components that describe the enhanced content referenced by the enhancements. The enhanced content may be modified using native applications for the enhanced content during editing of the program. The components describing the enhanced content in the editing system can be updated using identifiers of the enhanced content that are  
5 stored by the components. Enhanced content also may be automatically synchronized with time-based media.

Accordingly, in one aspect, in an editing system, a timeline is displayed that represents a program and associates enhanced content with a position in time on the timeline and associates time-based media with a position in time on the timeline. The  
10 timeline may include at least one track for enhanced content and at least one track for time-based media. A user interface allows a user to place a component representing enhanced content at a position in time on the timeline according to a selection of either a point in time or a duration on the timeline. The enhanced content represented by the component is associated with the point in time or the duration in the program according  
15 to the selection. An indication of the component is displayed on the timeline according to the selection.

In another aspect, an editing system displays a timeline representing a program for associating enhanced content with a position in time on the timeline and for associating time-based media with a position in time on the timeline. The timeline may  
20 include at least one track for enhanced content and at least one track for time-based media. A user is allowed to place a component representing enhanced content at a position in time on the timeline. The component representing enhanced content includes a field for storing a reference to a file defining the enhanced content. A user is allowed to place a component representing time-based media at a position in time on the timeline.  
25 The component representing time-based media includes a field for storing a reference to a file for storing the time-based media. The user also is allowed to perform editing operations on the timeline that affect temporal properties of components on the timeline. The editing operations process components representing enhanced content and components representing time-based media in the same manner.

30 In another aspect, an editing system for editing a program of time-based media and enhanced content has a bin for storing representations of content accessible by the

editing system for placement in the program. Enhanced content is imported into the bin of the editing system using an enhancement file. The enhancement file includes data defining one or more enhancements. An enhancement comprises attributes including an identifier of the enhancement and a reference to a file defining enhanced content  
5 associated with the enhancement. The enhancement file is processed to locate each enhancement. For each enhancement, a component is created in the bin of the editing system, wherein the component includes information describing the enhanced content.

In another aspect, a digital information product represents one or more enhancements describing enhanced content for placement in a program of time-based  
10 media and enhanced content using an editing system. This digital information product includes a computer readable medium and information stored on the computer readable medium that, when accessed by the computer is interpreted as an enhancement file. The enhancement files includes data defining one or more enhancements. An enhancement includes one or more attributes, including an identifier of the enhancement within the  
15 enhancement file, and a reference to a file defining enhanced content associated with the enhancement.

In another aspect, an editing system for editing a program of time-based media and enhanced content has at least one track for enhanced content and at least one track for time-based media. A document is generated in a markup language describing the at least  
20 one track for enhanced content by identifying each component representing enhanced content on the at least one track for enhanced content. For each identified component representing enhanced content, an element is created in the markup language document. This element includes a start position of the component on the at least one track for enhanced content, and, if the component is a source clip, then an end position of the  
25 component on the at least one track for enhanced content, and a reference to data in the markup language document defining an enhancement that describes the enhanced content represented by the component.

In another aspect, an editing system displays a timeline representing a program for associating enhanced content with a position in time on the timeline and for  
30 associating time-based media with a position in time on the timeline. The timeline may include at least one track for enhanced content and at least one track for time-based

media. A bin stores representations of enhanced content and time-based media accessible by the editing system for placement in the program. Enhanced content is imported into the bin to create one or more components in the bin. A component in the bin includes information describing the enhanced content including an identifier associated with the enhanced content. A user is allowed to place a component representing enhanced content at a position in time on the timeline, such that the enhanced content represented by the component is associated with the position in time in the program and an indication of the component is displayed on the timeline according to the position in time. The information describing the enhanced content related to a component is updated in the bin using the identifier associated with the enhanced content.

In another aspect, an editing system displays a timeline representing a program. The timeline may include at least one track for enhanced content and at least one track for time-based media. Using the timeline, enhanced content may be associated with a position in time on the timeline and time-based media may be associated with a position in time on the timeline. A user is allowed to relate enhanced content and time-based media to create auto-synchronized content. The user is allowed to place a component representing the auto-synchronized content at a position in time on the timeline. The enhanced content of the auto-synchronized content may be placed on the at least one enhanced content track and the time-based media of the auto-synchronized content may be placed on the at least one time-based media track. The auto-synchronized content represented by the component is associated with the position in time in the program. A user then can perform editing operations on the time-based media of the auto-synchronized content on the timeline such that the editing operations are also performed on the enhanced content of the auto-synchronized content.

#### BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is an illustration of a graphical user interface enabling editing of time-based media and enhanced content on a timeline.

Fig. 2A is an illustration of example data describing enhanced content.

Fig. 2B is an illustration of example data that may further describe enhanced content.

Fig. 3A is a flowchart describing how enhanced content may be imported into a bin.

Fig. 3B is a flowchart describing how interactive content may be imported into a bin.

Fig. 4 is a diagram illustrating a multi-user system for editing time-based media and enhanced content.

Fig. 5 is a flowchart describing how enhanced content may be refreshed.

5

## DETAILED DESCRIPTION

Time-based media, such as video, audio and animation, may be associated with a variety of types of enhanced content to create a time-based program in which enhanced content is temporally related to the time-based media.

10       Enhanced content includes any data that may be associated with a position in time in a time-based program. Data may represent, for example, an asset, a service or a function. Examples of such data include, but are not limited to, a database query, closed captioning information, a script of the dialogue of the program, subtitles or dubbing information, information from pre-production and production activities, cost or billing  
15 information related to the time-based media, rights management information, and interactive content.

Some enhanced content may be "opaque" to the editing system or to the program itself. Opaque content is content that uses an application other than the editing system for viewing the content, or that uses an application other than the editing system for editing  
20 the content, or that is not perceived during playback of the program in the editing system. Example uses of opaque content include supplying information for creating the program or tracking information about the program. Content that is opaque to the editing system may be encoded as part of the final program for distribution. Time-based media or nontemporal media that is not supported by the editing system, such as animation, may be  
25 represented as opaque content.

Interactive content includes, but is not limited to, documents defined in a markup language, documents of multiple media types, documents generated by the execution of a script or other computer program that is executed during the program, instructions or command signals sent to equipment, or other events or actions having a position in time  
30 during the program that result in content being perceived during playback of the program

in the editing system or on a playback device. Interactive content also may include nontemporal media data, such as a still image.

A kind of interactive content is called herein a "trigger." A trigger indicates an operation to be initiated at a point in time during playback of time-based media. Such operations may involve displaying pictures, graphics, images or other information, or other actions such as sending control signals to various devices. Control signals to equipment could be used in some applications, such as ride simulators. Information that may define a "trigger" is specified, for example, in the Advanced Television Enhancement Forum (ATVEF) specification for triggers, for example, as defined in section 1.1.5 of version 1.1 of the ATVEF specification. Other information specified by this and other interactive television formats may be used to define a trigger. A trigger also may indicate duration, synchronization information, or information about the size, position or orientation of display of time-based media associated with the trigger.

Fig. 1 illustrates an example user interface for a system for editing time-based media, such as video and audio, along with enhanced content to create a time-based program in which enhanced content is temporally related to the time-based media.

The user interface in Fig. 1 includes a source window 100 for displaying source media and a record window 102 for displaying an edited program. Video may be displayed in the record window 102 or in a region 104 of the record window 102. The region 104 may be specified by an indication of the size of the video with respect to the associated enhanced content, as described in more detail below, providing a picture-in-picture effect. The region 104 also may display video as a background on which associated enhanced content is overlaid. Other information also could be used to specify a layout of the video and the enhanced content to be viewed in the record window 102.

A timeline interface 106 represents the edited program, and includes one or more enhanced content tracks 112 and one or more time-based media tracks, such as one or more video tracks 108 and one or more audio tracks 110. A track has an associated edit rate or temporal resolution in which positions in time are represented. For example, all tracks may have the same edit rate, which may be determined by the edit rate of a primary video track.

The editing system may use representations of content, herein called "components," to allow a user to place content in the program being edited. A "bin" stores components that represent content that is accessible by the editing system for placement in the program being editing. A bin may be implemented using a folder of a file system of a general purpose computer. If a component representing content is placed at a position in time on a track in the timeline interface, the content represented by the component is associated with the position in time in the program and an indication of the component is displayed on the timeline at the position in time.

In general, to associate time-based media with a position in time on the timeline interface, a user may select from among components representing sources of time-based media from one or more bins. Time-based media represented by a selected component may be viewed in the source window 100. The user may select in and out points in the time-based media source viewed in the source window to designate a component, called a clip, which may be added to a sequence of clips on the time-based media track in the timeline interface 106.

To associate enhanced content with a position in time on the timeline interface, a user may select from among components representing enhanced content from the one or more bins. Information associated with a selected component may be viewed in the source window 100 or in a separate viewing application for the enhanced content, depending on the type of the enhanced content represented by the component and the capabilities of the editing system. The user may select an enhanced content track 112 and a position in time in the track at which the component should be added. The component may be added at a point in time, using a component called a locator 114 (described below), or may be added over a range of time or duration, using a component called a source clip 116 (described below). Time-based media also may be represented using source clips and locators. A user may be prompted by the editing system for a selection of either a point in time or a duration to which the enhanced content is associated.

The timeline interface may be implemented to enable components representing enhanced content to be placed on the same track on the display as the time-based media. Such components may be, for example, either clips or locators. Separate data structures may be used to represent a collection of the components representing the enhanced

content and a collection of the components representing the time-based media that are on the same track in the timeline display.

A source clip references a clip in a bin and has a start position and duration in the track. The duration may be represented by a duration or a stop time on the enhancement  
5 track. A source clip also may have attributes. The source clip may include a field indicating the type of data that it represents, such as video or audio.

A locator is attached to a clip or other component in the timeline at a point in time on the clip. A locator also may have attributes. The locator may include a field  
10 indicating the type of data that it represents, such as interactive content or opaque enhanced content.

Because the types of components used for time-based media and enhanced content are the same, enhanced content has the same behavior as the time-based media. This behavior can be achieved in several ways. For example, components representing time-based media and components representing enhanced content may be instances of  
15 subclasses of, and that inherit editing operations from, an object class for which the editing operations are implemented. Components representing time-based media and components representing enhanced content also may be instances of the same object class for which the editing operations are implemented. Components representing time-based media and components representing enhanced content also may be represented using data  
20 structures that are acted upon by procedures that implement the editing operations and that provide the same results for these editing operations regardless of the type of the content represented by the component. Thus, editing operations such as cut, slip, slide, trim, lift, splice and overwrite and auto-synchronization of a component representing enhanced content with another source, may be used to edit both time-based media and  
25 enhanced content on the timeline and maintain frame accurate synchronization between the enhanced content and the time-based media. Such editing operations are described in more detail in U.S. Patent 5,754,851 and 6,118,444, which are hereby incorporated by reference.

An enhanced content track in the timeline interface may allow a user to place both  
30 source clips and locators on the enhanced content track. One or more of such tracks may be provided. Alternatively, separate tracks may be provided for source clips and locators.

Separate data structures may be used to represent a track for source clips that represent enhanced content and a track for locators that represent enhanced content.

The editing system also may be configured so that editing of the time-based media is disabled, allowing an editor to focus primarily on the relationship of the enhanced content on the enhanced content track and with the time-based media, if any. Such an editing system can export the edited program directly to another editing system that is enabled to edit both the enhanced content and the time-based media.

An edited program may be represented using any of several data structures, which may be stored in any of a number of formats. For example, a system may use structures corresponding to the Advanced Authoring Format (AAF) specification, Open Media Framework (OMF) specification, or structures described in U.S. Patents 6,061,758 and 5,754,851, which are hereby incorporated by reference. In general, the data structure representing the edited program allows a track to be represented as a collection of temporally related components. These components may be represented using collections of sequential components and collections of concurrent components. Kinds of clips may include clips that reference time-based media and enhanced content, of which there are several types described in more detail below. Using AAF, a clip representing enhanced content may be represented as a master clip having the data definition "META." The information associated with the clip may be stored as an attribute sub-list called a "\_META " list that includes type-value pairs. An attribute may be stored as a string type. Using AAF, a locator representing enhanced content may be represented as a locator with an attribute list called "\_META ". It also may have a data definition of "META."

To allow editing of enhanced content into a program, enhanced content may be imported into bins to create components that may be accessed by the editing system. In a bin, the enhanced content is represented by a clip or other component, with information describing the enhanced content stored as an attribute of the clip or other component. An attribute is, in general, a data field that is used to store a variety of user defined data. To simplify importing enhanced content into a bin, an input format for describing enhanced content is provided. This input format and the process of importing will now be described in more detail.

Information describing enhanced content may be stored as a data file, herein called an "enhancement file," which may be stored in a directory in a file system in either local or shared storage or as a resource accessible through a computer network. An enhancement file includes data defining one or more enhancements that describe enhanced content. The enhancement file may be represented using a markup language such as the eXtensible Markup Language (XML) according to a document type definition (DTD) for enhancement files. An example of such a DTD is provided in Appendix A.

In the example shown in Appendix A, an enhancement is defined by one or more attributes (called a "core attribute list" in Appendix A) and one or more custom attributes (called a "custom attribute list" in Appendix A). The one or more attributes includes the identifier ("id") assigned to the enhancement and a "name" for the enhancement. The identifier of the enhancement is unique within the enhancement file. A type of the enhancement is provided to allow a system to process enhancements of different types in different ways. An example type may be "interactive content" (or "ITV"), or "opaque".

An "opaque" type of enhancement is an enhancement that represents opaque content. The type also may be further described by a "sub-type." A "MIME-type" field indicates the MIME type of the data of the enhancement. A locator field indicates a file that defines the enhanced content associated with the enhancement. A native application field indicates an application that can at least open and read the file specified in the locator field. Other tracking information such as the creator, creation date, modifiers and dates of modification also may be included. One or more custom attributes also may be added by a user.

An enhancement file also may specify an ITV type of enhancement, as described above. If an enhancement file and editing system are limited to processing ITV type enhanced content, such as triggers, the enhancement file format may be specified using a document type definition (DTD) specialized for this type of content. An example of such a DTD is provided in Appendix B. In this example format, an enhancement file includes a list of triggers, and a trigger is defined by a locator ("url") that references a file associated with the trigger, and a "name" for the trigger. Other information indicates a date and time at which the trigger "expires." A "script" designates computer program code that may be executed by or for the trigger. An identifier field ("id") stores an

identifier for the trigger in the enhancement file. A "checksum" and additional field for "user data" also may be included.

The information describing enhanced content from an enhancement file may be stored by a component in a bin by importing the enhancement file into an editing system.

5 A component in the bin may include information such as shown in Fig. 2A. In particular, an indication of a type of the enhanced content 200 may be provided if numerous types of enhanced content are supported. A locator 201, such as a uniform resource locator (URL) or other file name, may be used to indicate a file that defines the enhanced content. An indication of an application 202, which may be a script for a trigger, such as  
10 a Java script program or other computer program code to be executed, may be included. A name 204 also may be stored. This name field may be used as a name or as a readable text description of the enhanced content. Expiration information 206, such as a date and time, and expiration type 208 (indicating whether the expiration information indicates a duration of time the trigger is valid or a time at which the trigger expires) also may be  
15 provided for triggers. An identifier field 210 stores the identifier of the enhancement that describes the enhanced content, which may include a reference to the enhancement file. One or more additional fields 212 may be used to store other data such as user defined data, custom attributes, modification and creation dates and user information, or other information described by an enhancement in the enhancement file. A checksum 214 may  
20 be included to allow detection of corrupted data.

If the enhanced content is defined by a file for which an image can be created, such as a URL that references an HTML document, information such as shown in Fig. 2B may describe the enhanced content further. This information may include a snapshot 220 of the file. For example, if the file is a document retrieved from a URL, the document  
25 may be processed by a browser to generate a snapshot. Any related files 222 (such as linked files), and any indication of cropping 224, scaling 226 or overlay 228 of the associated video on the video track, and the dimensions and position 230 of the video with respect to the snapshot also may be stored. The dimensions 232 of the snapshot also may be obtained and stored. Whether the snapshot is to be overlaid on the video or  
30 whether the video is to be displayed within the snapshot as a picture in picture or other layout information 234 also may be specified. The information shown in Fig. 2B also is

an example of additional kinds of information that may be used to represent interactive content, such as a document in a markup language such as the hypertext markup language (HTML) or XML.

There are several ways to create enhancement files. For example, an  
5 enhancement application may be used to create and modify enhancement files. The enhancement application may be any application that may be used to generate an XML file or file in other suitable format, such as a character delimited file, that may be used to specify fields and associated data for those fields. A spreadsheet application or word processing application, for example, may be used. Using an XML file that conforms to a  
10 document type definition allows the format of each enhancement file to be validated.

The enhancement application also may be used to assign a unique identifier (UID) to each enhancement in the enhancement file, or such UIDs may be assigned manually to each enhancement. The UID for an enhancement is stored in the enhancement file. The UID need not be a global unique identifier (GUID), but may be unique only within the  
15 enhancement file. The enhancement application also may receive an XML file defining an enhancement file without UIDs and add UIDs to create an enhancement file with the UIDs.

The process of importing enhanced content into a bin using an enhancement file will now be described in connection with Fig. 3A. A user first identifies 300 an  
20 enhancement file using any conventional technique to locate the file. An import operation is then invoked 302. The import operation reads 304 the enhancement file to access the data defining one or more enhancements in the enhancement file. After data defining the enhancement is read, a component is created 306 in the bin, with attributes that store information describing the enhanced content. How the data defining the  
25 enhancement is used to create a component in the bin may depend on the type of the enhanced content. The attributes that are stored may include the name of the enhancement file storing the enhancement and the identifier of the enhancement in that enhancement file. The attributes that are stored may include a reference to the file defining the enhanced content associated with the enhancement. A reference to a native  
30 application for the file also may be stored. A component so created also may have its

own unique identifier assigned by the editing application, which is different from the identifier of the corresponding enhancement.

How an enhancement is processed to define a component in the bin may depend on the type of the enhanced content. As an example, if the enhanced content is  
5 interactive content, and more particularly a trigger, then (referring now to Fig. 3B) the enhancement is located 310 in the enhancement file. The file defining the trigger that is referenced by the enhancement is accessed 312 to obtain the information described in Fig. 2B. In particular, any "TV:" element or other element in the specified file that specifies the dimension and position of the video and its relationship to the enhanced  
10 content and is read to obtain appropriate values. A snapshot of the document defined by the specified file may be generated 314 by applying the document to a conventional browser object and storing the output from the browser object as an image file. The browser object may be instructed to use a background of a unique color, such as green or blue as is commonly used in blue screen and green screen special effects to allow the  
15 HTML data to be overlaid on the video using a chroma key operation. A reference to the snapshot may be stored 316 as an attribute of the component in the bin.

It is possible to access and process the entire specified file and files referenced within the specified file for later use. Referenced files might be imported to protect against subsequent unavailability of the specified file or referenced files. Whether the  
20 import process includes capturing of HTML data referenced by the URL 200, or the files referred to by the document at the URL 200 may be at the user's selection through an appropriate user interface. It is also possible to import only the HTML data (or other information) for interactive content that is not a trigger. For example, a document in a markup language may be accessed through conventional techniques for locating its file.  
25 Files accessed in this manner include any files referenced by a file referenced by a trigger. The file then may be read to extract information that is stored as an attribute of a component in the bin.

Thus, if a trigger is imported into a bin, a component in the bin stores a unique reference for the trigger by storing a reference to the enhancement file and the identifier  
30 of the trigger in that enhancement file. The component in the bin also captures the information defining the trigger from the enhancement file and optionally other

information from files associated with the trigger, such as the information shown in Figs. 2A and 2B. For example, if the trigger includes a reference to information to be displayed with the video data or about the display of the video data, this information also may be extracted and stored with the information about the trigger in the bin. Similarly, a document in a markup language, or other interactive content, may be processed to extract information about the document, such as the information described above in connection with Fig. 2B. This information and the reference to the enhancement file are stored as an attribute of a clip or other component. Thus, kinds of components representing triggers in a bin may include HTML clips that reference hypertext markup language (HTML) (or other markup language) data, trigger clips that reference information about triggers, and linked trigger clips that reference both information about triggers and HTML (or other markup language) content. The type of a trigger (whether HTML, Trigger or Linked Trigger) depends on the attributes associated with it.

After enhanced content is imported into a bin, a user may access components representing the enhanced content in the bin using the editing application. Components representing time-based media also may be in a bin. One or more bins may be provided. Separate bins may be provided for different types of content, or a bin may include content of mixed types. Components in a bin may be selected, optionally viewed in a source viewer depending on the type, and added to a timeline. Addition of components to a timeline may be limited to a designated track (e.g., enhanced content may be limited to the enhanced content track 112 in Fig. 1). On an enhanced content track, enhanced content may be added as one of a number of types of components, such as a source clip or a locator. Enhanced content may be added as a locator on tracks other than the enhanced content track. The information describing the enhanced content is stored as attributes on either the source clip or locator. For triggers, a trigger clip in a bin appears as a locator on the timeline, and the attributes of the trigger clip are transferred to the locator. An HTML clip in a bin appears as a source clip on the timeline, and its attributes are transferred to the source clip. A linked trigger clip and other components representing enhanced content in a bin may appear, upon a user's selection, as either a source clip or a locator on a timeline. The user's selection may be obtained through any appropriate user interface.

Components representing enhanced content also may be automatically synchronized to components representing time-based media in the bin by associating the components with each other. This association creates a new component that may be added to a program. With the new component, the component representing enhanced  
5 content remains associated with the component representing time-based media through operations performed on the time-based media during editing. The editing operations performed in the time-based media also are performed on the associated component representing enhanced content.

Because the types of components used for time-based media and enhanced  
10 content are the same, enhanced content has the same behavior as the time-based media in response to editing operations. Thus, editing operations such as cut, slip, slide, trim, lift, splice and overwrite and auto-synchronization of the enhanced content with another source, may be used to edit both time-based media and enhanced content together and maintain frame accurate synchronization between the enhanced content and the time-  
15 based media.

If a program has been edited using the timeline interface described above, it may be played back, for example in the record window of Fig.1 or to an external monitor. If the some enhanced content includes information, such as shown in Fig. 2B, that indicates how the video should be displayed with the enhanced content, then this information may  
20 be used to control the display of the video at the position in time for that enhanced content.

Referring now to Fig. 4, an example system for simultaneous authoring of time-based and enhanced content for a program is described. In this example, enhancement files 408 may be created and modified by the enhancement application 402, described  
25 above, and stored on shared storage 400. Such shared storage may be a computer network, a shared file system, a distributed file system, a set of file servers or any other mechanism to share information between the systems. The enhancement application 402 uses a reference to the enhancement file, such as the file name, and an identifier 406 of an enhancement to access the shared storage for reading and/or writing of data defining  
30 enhancements in enhancement files 408. Enhancements may refer to content files 410 and 412 that are stored on the shared storage 400 or some other location. These content

files may be created and modified using a variety of native applications 414 and 416. The native applications may be, for example, content authoring tools, or a database application, a script writing program, or any other application that creates a file defining enhanced content. An editing application 404, such as described above in connection  
5 with Figs 1-3, may be used to create and modify the program by combining time-based media and enhanced content in the manner described above. Enhanced content 418 may be imported into a bin for the editing application 404 using enhancement files 408 and content files 410, 412, to create components that may be selected for insertion into the program. An identifier 420 for the enhanced content 418, such as a reference to the  
10 enhancement file 408 and an identifier for the enhancement describing the enhanced content, or other identifier, such as a reference to a file defining the enhanced content, may be used by the editing application to access the enhanced content 418.

The editing application also may be programmed to launch a native application 414, 416 associated with an enhanced content that is stored in a bin or that is placed on an  
15 enhanced content track. For example, the editing application may be programmed to allow a user to use any conventional operation to select the enhanced content that is in the bin or on the enhanced content track. The editing application then can cause the associated native application to launch, access and open for editing the enhanced content associated with the component that is in the bin or on the enhanced content track.

20 With such a system, multiple editors may be working on different parts of a program at one time. Thus, enhanced content files 410 and 412 or enhancement files, or enhancements within them, may change after they are imported into a bin of the editing application 404. However, the unique references for the enhanced content in the bin, e.g., the enhancement file name and identifier for an enhancement or a reference to a file  
25 defining the enhanced content, may be used to obtain updated enhanced content from its source. This updating process is called a refresh operation. The refresh operation is similar to an import operation except for the method of identification of the enhancement file. In a refresh operation, information describing the enhanced content that has been selected for refresh is extracted from currently available sources that correspond to the  
30 identifiers associated with the enhanced content, e.g., the identifier for an enhancement or the reference to a file defining the enhanced content.

Referring to Fig. 5, to perform a refresh operation on enhanced content, the user may select 500 one or more components in the bin to be refreshed, for example by selecting a particular component, components that reference the same identifier, or components on a track or components in a bin. One of the selected components is  
5 selected 502. The name of the enhancement file and the identifier of the enhancement for the selected components are used to locate 504 the enhancement file from the shared storage (400 in Fig. 4). The enhancement is then imported 506 in the same manner as described above in connection with Figs. 3A and 3B. If no components remain to be refreshed, as determined in 508, the refresh operation is complete, otherwise, the next  
10 component of the selected components is then selected 502 and the steps 502-508 are repeated. Similar operations also can be performed on other enhanced content using an identifier, e.g., a URL or other reference to a file that defines the enhanced content.

Upon completion of the editing of a program that includes both enhanced content and time-based media, it is possible that there are many possible distribution formats for  
15 the program. Therefore, the program may be transformed from its representation in the editing system, using the program data structures and enhanced content, into one or more encoded distribution formats, such as ATVEF, WebTV, Liberate, broadband interactive TV, wireless, digital versatile disk (DVD), or Internet or computer streaming format or other format for the particular distribution channel or interactive application, using  
20 encoding techniques appropriate to the application.

To simplify the encoding process the editing system may output a file describing the program and the enhanced content. The output may be, for example, a file conforming to the AAF specification. Such a file also may be used to generate an XML file that describes the content of the enhanced content track(s) of the program. Such a  
25 file may conform to a document type definition as shown in Appendix C. This DTD includes the same information as in the DTD in Appendix A, but also defines an "interactive timeline" element that include an AAF header element, an edit rate element, a start time element and one or more meta track elements. The AAF header defines information relevant to the associated AAF file that can be created by the editing system.  
30 The edit rate element indicates the temporal resolution of the editing operations for the program. The start time represents the time code of the first frame of the program. A

metatrack element is represented by a track number, a track name, a reference to a track that includes clips and an optional reference to a track that includes locators, if a separate tracks are used for representing clips and locators of enhanced content.

5 An enhanced content track for clips is represented by an integer "slotid" that is the identifier in AAF of the track, and one or more clips. A clip is represented by the element in the markup language document defining the associated enhancement and data representing the start and end times of the corresponding enhanced content on the track. The "startframes" and "endframes" refers to the start time and end time of the clip on the track in frames. The data representing the start and end times of the clip on the track also  
10 may be represented in any of a number of timecode formats, including, but not limited to, 24 frame per second, 25 frames per second with pulldown, 25 frames per second, 30 frames per second in drop frame format, and 30 frames per second in non-drop frame format. 23.976 frame per second format also may be provided. If the edit rate is 24 frames per second, the start and end times can be readily maintained in all of the different  
15 formats.

An enhanced content track for locators, if a separate track is maintained, is represented by an integer "slotid" that is the identifier in AAF of the track, and one or more locators. A locator is represented by the element in the markup language document defining the associated enhancement, and data representing the position of the enhanced  
20 content on the track. The "positionframes" refers to the position of the locator on the track in frames. The data representing the position of the locator on the track may be represented in any of a number of timecode formats, including, but not limited to, 24 frame per second, 25 frames per second with pulldown, 25 frames per second, 30 frames per second in drop frame format, and 30 frames per second in non-drop frame format.  
25 23.976 frame per second format also may be provided. If the edit rate is 24 frames per second, the position can be readily maintained in all of the different formats.

By using enhanced content in the manner described above, various information can be associated with time-based media during editing.

As one example, information about the source, cost or usage rights associated  
30 with time-based media used in the program may be stored for tracking, billing or compliance services. The components representing enhanced content may be

automatically synchronized with the time-based media in the bin. The actual usage of the time-based media in the program, in connection with this stored information, can be used for billing, tracking and compliance regarding media rights.

As another example, a scriptwriting program can be used to create a list of characters associated with a film. A character name may be associated with the lines for the character in the dialog. The information about a character may be referenced by a component that also references back to the screenplay. A clip of time-based media may be associated with a component that references a character in that clip. Through this association between the screenplay and the time-based media, it can be determined when each character spoke, how long each character spoke, and what each character said. Similarly, the script information can be used to assist speech recognition of the audio track, identify time-based media clips in a bin that are associated with the script, provide subtitles or closed captioning or assist in dubbing.

As another example, information about primary scene locations in a program may be placed in components that are automatically synchronized with the time-based media for the scene through an association in the bin. This information in the components could be used for DVD authoring, for example, by providing information to create an index of the program.

As another example, instructions to a ride simulator, such as up, down, left, right, skew left and skew right or other commands to various mechanical devices used in such entertainment venues, may be placed in components. Such components may be placed on the enhanced content track to represent when, in association with the time-based media, the ride simulator or other device takes the corresponding actions. The instructions to the device may be in a format that can be processed by the device, such as a GPI trigger for a ride simulator. For some ride simulator devices, the motion of the device is represented by a start and stop position of the device. The speed at which the motion is made is a function of time, which may be represented on the enhanced content track by representing the motion as a source clip. The duration of the clip on the enhanced content track may be used to represent the time in which the device moves from the start to the stop position, and thus determines the speed of motion. The instructions to the device also may be converted from a format specified in the

component into another format in which the information can be transmitted to and processed by the device. For example, instructions may be encoded in a communication format such as infrared signals, audio signals out of the audible range, other wireless interconnect or network communication protocol.

5       As another example, the time-based media may be associated with environmental and lighting controls represented by enhanced content on the enhanced content track. A component may represent instructions to open or close curtains and dim or brighten lights in a theatre, for example.

10       As another example, the components representing enhanced content may be used to represent images, drawings or other information for storyboarding. Various components may be created to reference files that represent scenes of the program. Different types of files could be used to represent different scenes. These components could be placed on the enhanced content track as source clips, to signify the timing of the scenes in the storyboard. This editing of a storyboard using the enhanced content track  
15       could be performed before the time-based media is available for editing. Whether playback of the storyboard in the editing system could be performed depends on the availability of the native applications or other application to convert the referenced files into image and/or sound data that can be processed by the editing system.

20       Such a system may be implemented using a computer program on a general purpose computer. Such a computer system typically includes a processor, an input device, a display device, and a memory. The memory stores software for performing the various functions described above. The computer display device displays a software generated user interface such as shown in Fig. 1 to accommodate the functionality.

25       The computer system may be a general purpose computer which is available from a number of computer system manufacturers as is well known by those of ordinary skill in the art. The computer system executes an operating system, such as Windows by Microsoft Corporation, MAC OS X by Apple Computer, Solaris by Sun Microsystems, Inc., IRIX by Silicon Graphics, Inc., or a version of UNIX. The invention is not limited to any particular computer system or operating system or programming language or  
30       programming techniques. The memory stores data and instructions. The memory may include both a volatile memory such as RAM and non-volatile memory such as a ROM, a

magnetic disk, an optical disk, a CD-ROM or the like. The input device allows the user to interact with the computer system. The input device may include, for example, one or more of a keyboard, a mouse, or a trackball. The display device displays a user interface. The display device may include, for example, a cathode ray tube (CRT), a flat panel display, or some other display device.

Having now described example implementations, it should be apparent to those skilled in the art that the foregoing is merely illustrative and not limiting, having been presented by way of example only. Numerous modifications and other implementations are within the scope of one of ordinary skill in the art and are contemplated as falling within the scope of the invention.

What is claimed is:

## APPENDIX A

```

<?xml version="1.0" encoding="UTF-8"?>

<!ENTITY % STRING          "CDATA #REQUIRED">
<!ENTITY % TYPE             "type %STRING; ">
<!ENTITY % VALUE            "value %STRING;">

<!ELEMENT enhancements      (enhancement)+ >
<!ATTLIST enhancements     version %STRING;>
<!ELEMENT enhancement
      (core_attribute_list,custom_attribute_list*)>

<!ELEMENT core_attribute_list (id, name, type, sub-type,
MIME-type, locator, native, created_by, created_on,
modified_by, modified_on, modified_from)>

<!ELEMENT id EMPTY>
<!ATTLIST id type CDATA #FIXED "ID" %VALUE;>

<!ELEMENT name EMPTY>
<!ATTLIST name type CDATA #FIXED "META Name" %VALUE;>

<!ELEMENT type EMPTY>
<!ATTLIST type type CDATA #FIXED "META Type" %VALUE;>

<!ELEMENT sub-type EMPTY>
<!ATTLIST sub-type type CDATA #FIXED "META Sub-type"
%VALUE;>

<!ELEMENT MIME-type EMPTY>
<!ATTLIST MIME-type type CDATA #FIXED "META Type"
%VALUE;>

<!ELEMENT locator EMPTY>
<!ATTLIST locator type CDATA #FIXED "File Location"
%VALUE;>

<!ELEMENT native EMPTY>
<!ATTLIST native type CDATA #FIXED "Native Application"
%VALUE;>

<!ELEMENT created_by EMPTY>
<!ATTLIST created_by type CDATA #FIXED "Created by"
%VALUE;>

```

```
<!ELEMENT created_on EMPTY>
<!ATTLIST created_on      type CDATA #FIXED "Created on"
%VALUE;>

<!ELEMENT modified_by EMPTY>
<!ATTLIST modified_by type CDATA #FIXED "Modified by"
%VALUE;>

<!ELEMENT modified_on EMPTY>
<!ATTLIST modified_on type CDATA #FIXED "Modified on"
%VALUE;>

<!ELEMENT modified_from EMPTY>
<!ATTLIST modified_from type CDATA #FIXED "Modified from"
%VALUE;>

<!ELEMENT custom_attribute_list(attribute)*>

<!ELEMENT attribute EMPTY>
<!ATTLIST attribute %TYPE; %VALUE;>
```

## APPENDIX B

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
    !! The trigger-list includes none, one, or many
    triggers
    !! A trigger is represented by
    !! <URL> [attr1 : val1][attr2 : val2] ...[attrn :
    valn][checksum]
-->
<!--ELEMENT trigger-list (trigger)*>
<!--ELEMENT trigger ((url) | (name)? | (expires)? | (script)?
| (checksum)? | (user-data)?)>
<!--ELEMENT url (#PCDATA)>
<!--ELEMENT name (#PCDATA)>
<!--ELEMENT expires ((date)? | (time)?)>
<!--ELEMENT date (year, month, day)>
<!--ELEMENT year (#PCDATA)>
<!--ELEMENT month (#PCDATA)>
<!--ELEMENT day (#PCDATA)>
<!--ELEMENT time (hours, minutes, (seconds)?)>
<!--ELEMENT hours (#PCDATA)>
<!--ELEMENT minutes (#PCDATA)>
<!--ELEMENT seconds (#PCDATA)>
<!--ELEMENT script (#PCDATA)>
<!--ELEMENT id (#PCDATA)>
<!--ELEMENT checksum (#PCDATA)>

```

## APPENDIX C

```
<!ELEMENT enhancements (enhancement)+>
<!ATTLIST enhancements version CDATA #REQUIRED>

<!ELEMENT enhancement (core_attribute_list,
custom_attribute_list*)>

<!ELEMENT core_attribute_list (id, name, type, sub-type,
MIME-type, locator, native, created_by, created_on,
modified_by, modified_on, modified_from)>

<!ELEMENT id EMPTY>
<!ATTLIST id type CDATA #FIXED "ID" value CDATA #REQUIRED>

<!ELEMENT name EMPTY>
<!ATTLIST name type CDATA #FIXED "META Name" value CDATA
#REQUIRED>

<!ELEMENT type EMPTY>
<!ATTLIST type type CDATA #FIXED "META Type" value CDATA
#REQUIRED>

<!ELEMENT sub-type EMPTY>
<!ATTLIST sub-type type CDATA #FIXED "META Sub-type" value
CDATA #REQUIRED>

<!ELEMENT MIME-type EMPTY>
<!ATTLIST MIME-type type CDATA #FIXED "META Type" value
CDATA #REQUIRED>

<!ELEMENT locator EMPTY>
<!ATTLIST locator type CDATA #FIXED "File Location" value
CDATA #REQUIRED>

<!ELEMENT native EMPTY>
<!ATTLIST native type CDATA #FIXED "Native Application"
value CDATA #REQUIRED>

<!ELEMENT created_by EMPTY>
<!ATTLIST created_by type CDATA #FIXED "Created by" value
CDATA #REQUIRED>

<!ELEMENT created_on EMPTY>
<!ATTLIST created_on type CDATA #FIXED "Created on" value
CDATA #REQUIRED>
```

```

<!ELEMENT modified_by EMPTY>
<!ATTLIST modified_by type CDATA #FIXED "Modified by" value
CDATA #REQUIRED>

<!ELEMENT modified_on EMPTY>
<!ATTLIST modified_on type CDATA #FIXED "Modified on" value
CDATA #REQUIRED>

<!ELEMENT modified_from EMPTY>
<!ATTLIST modified_from type CDATA #FIXED "Modified from"
value CDATA #REQUIRED>

<!ELEMENT custom_attribute_list (attribute)*>

<!ELEMENT attribute EMPTY>
<!ATTLIST attribute type CDATA #REQUIRED value CDATA
#REQUIRED>

<!-- This is the root tag of the XML file. It defines the
Interactive Timeline. -->

<!ELEMENT InteractiveTimeline (AAFHeader, editrate,
starttime, META_Track*)>

<!ATTLIST InteractiveTimeline version CDATA #REQUIRED>

<!-- Header provides file-wide information and indexes. An
AAF file shall have one and only one Header object -->
<!ELEMENT AAFHeader (lastmodified, version,
generationaid+)>

<!-- Time and date the file was last modified-->
<!ELEMENT lastmodified (#PCDATA)>

<!-- Version number of the document that the file conforms
to; shall be 1.0 or higher-->
<!ELEMENT version (#PCDATA)>

<!-- AUID generated at the time the application created or
opened file for modification -->
<!ELEMENT generationaid (#PCDATA)>
<!-- This is the END of the AAF Header -->

<!-- This is the Composition EditRate -->
<!ELEMENT editrate (#PCDATA)>

<!-- This is the Composition StartTime in TimeCode -->

```

```

<!ELEMENT starttime (#PCDATA)>

<!-- The InteractiveTimeline consists of one or more
META_Tracks -->
<!ELEMENT META_Track (physicaltracknumber,
physicaltrackname, timelinemobslot, eventmobslot?)>

<!-- Specifies the physical channel -->
<!ELEMENT physicaltracknumber (#PCDATA)>

<!-- Specifies an optional text name for the slot -->
<!ELEMENT physicaltrackname (#PCDATA)>

<!-- Specifies the timeline mob slot which can contain 0 or
more enhancementclips-->
<!ELEMENT timelinemobslot (slotid, enhancementclip*)>

<!-- Specifies an integer that is used to reference the mob
slot -->
<!ELEMENT slotid (#PCDATA)>

<!-- Specifies an enhancementclip which contains an
enhancement and its start and end times on the track. The
start and end times will always be specified in frames.
Depending on the project start and end times in additional
timecodes will be available -->

<!ELEMENT enhancementclip (enhancement, starttime_frames,
endtime_frames, starttime_24?, endtime_24?, starttime_25P?,
endtime_25P?, starttime_25?, endtime_25?, starttime_30D?,
endtime_30D?, starttime_30ND?, endtime_30ND?)>
    <!ELEMENT starttime_frames (#PCDATA)>
    <!ELEMENT starttime_24 (#PCDATA)>
    <!ELEMENT starttime_25P (#PCDATA)>
    <!ELEMENT starttime_25 (#PCDATA)>
    <!ELEMENT starttime_30D (#PCDATA)>
    <!ELEMENT starttime_30ND (#PCDATA)>
    <!ELEMENT endtime_frames (#PCDATA)>
    <!ELEMENT endtime_24 (#PCDATA)>
    <!ELEMENT endtime_25P (#PCDATA)>
    <!ELEMENT endtime_25 (#PCDATA)>
    <!ELEMENT endtime_30D (#PCDATA)>
    <!ELEMENT endtime_30ND (#PCDATA)>

<!-- Specifies the event mob slot which can contain 1 or
more enhancementlocators-->
<!ELEMENT eventmobslot (slotid, enhancementlocator+)>

```

<!-- Specifies an enhancementlocator which contains an enhancement and its position on the track. The position will always be specified in frames. Depending on the project position in additional timecodes will be available -->

<!ELEMENT enhancementlocator (enhancement, position\_frames, position\_24?, position\_25P?, position\_25?, position\_30D?, position\_30ND?)>

<!ELEMENT position\_frames (#PCDATA)>

<!ELEMENT position\_24 (#PCDATA)>

<!ELEMENT position\_25P (#PCDATA)>

<!ELEMENT position\_25 (#PCDATA)>

<!ELEMENT position\_30D (#PCDATA)>

<!ELEMENT position\_30ND (#PCDATA)>

## CLAIMS

1. An editing system comprising:

a timeline interface for displaying a timeline representing a program and for associating enhanced content with a position in time on the timeline and for associating  
5 time-based media with a position in time on the timeline; and

a user interface for allowing a user to place a component representing enhanced content at a position in time on the timeline according to a selection of either a point in time or a duration on the timeline, such that the enhanced content represented by the component is associated with the point in time or the duration in the program according  
10 to the selection and an indication of the component is displayed on the timeline according to the selection.

2. The editing system of claim 1, wherein the timeline includes at least one track for enhanced content and at least one track for time-based media.

3. The editing system of claim 1, wherein the component representing enhanced content is placed on the timeline at a point in time using a locator component.

4. The editing system of claim 1, wherein the component representing enhanced content  
20 is placed on the timeline with a duration using a source clip component.

5. The editing system of claim 1, wherein the component representing enhanced content on the timeline includes a field for storing a reference to a file defining the enhanced content.

6. The editing system of claim 5, wherein the component representing enhanced content on the timeline includes a field for storing a reference to a native application capable of processing the enhanced content.

7. The editing system of claim 5, wherein a native application capable of processing the enhanced content is determined according to the file defining the enhanced content.

8. The editing system of claim 1, further comprising:

means for allowing a user to place a component representing time-based media at a position on the timeline, wherein the component has a duration.

5

9. The editing system of claim 8, wherein the component representing time-based media is placed on the timeline with the duration using a source clip component.

10. The editing system of claim 9, wherein the component representing enhanced content is placed on the timeline with a duration using a source clip component.

10

11. The editing system of claim 8, wherein the component representing time-based media on the timeline includes a reference to a file for storing the time-based media and a range within the stored time-based media.

15

12. The editing system of claim 1, further comprising:

means for allowing a user to place a component representing time-based media at a position on the timeline, wherein the component has a point in time on the timeline.

13. The editing system of claim 12, wherein the component representing time-based media is placed on the timeline at the point in time using a locator component.

20

14. The editing system of claim 13, wherein the component representing enhanced content is placed on the timeline at a point in time using a locator component.

25

15. The editing system of claim 1, wherein the means for allowing includes means for prompting a user for the selection.

16. The editing system of claim 1, wherein enhanced content comprises nontemporal media data.

30

17. The editing system of claim 1, wherein enhanced content comprises data that is not processed natively by the editing system.

18. The editing system of claim 1, wherein enhanced content comprises interactive  
5 content having a time during the program that results in content being perceived during playback of the program.

19. The editing system of claim 18, wherein the interactive content comprises a document defined in a markup language.

10

20. The editing system of claim 19, wherein the document is generated by the execution of a computer program that is executed during the program

15

21. The editing system of claim 18, wherein the interactive content comprises instructions or command signals sent to equipment

22. The editing system of claim 18, wherein the interactive content comprises a trigger that stores an indication of an operation to be initiated at a point in time during playback of time-based media.

20

23. The editing system of claim 22, wherein the operation comprises displaying data.

24. The editing system of claim 22, wherein the operation comprises sending control signals to a device.

25

25. The editing system of claim 22, wherein the trigger also comprises information about the size, position or orientation of display of time-based media associated with the trigger.

30

26. The editing system of claim 1, wherein enhanced content comprises data that is opaque to the editing system.

27. The editing system of claim 26, wherein the opaque enhanced content uses an application other than the editing system for viewing the enhanced content.

5 28. The editing system of claim 26, wherein the opaque enhanced content uses an application other than the editing system for editing the enhanced content.

29. The editing system of claim 26, wherein the opaque enhanced content is not perceived as part of the program by the user during playback of the program in the  
10 editing system.

30. The editing system of claim 26, wherein the opaque enhanced content supplies information to create the program.

15 31. The editing system of claim 26, wherein the opaque enhanced content tracks information about the program.

32. A method for editing a time-based program of time-based media and enhanced content, comprising:

20 displaying a timeline representing a program and for associating enhanced content with a position in time on the timeline and for associating time-based media with a position in time on the timeline; and

allowing a user to place a component representing enhanced content at a position in time on the timeline according to a selection of either a point in time or a duration on  
25 the timeline, such that the enhanced content represented by the component is associated with the point in time or the duration in the program according to the selection and an indication of the component is displayed on the timeline according to the selection.

33. A computer program product, comprising:

30 a computer-readable medium;

computer program instructions stored on the computer readable medium that, when executed by a computer, instruct the computer to perform a method for editing a time-based program of time-based media and enhanced content, comprising:

displaying a timeline representing a program and for associating enhanced content  
5 with a position in time on the timeline and for associating time-based media with a position in time on the timeline; and

allowing a user to place a component representing enhanced content at a position in time on the timeline according to a selection of either a point in time or a duration on the timeline, such that the enhanced content represented by the component is associated  
10 with the point in time or the duration in the program according to the selection and an indication of the component is displayed on the timeline according to the selection.

34. An editing system comprising:

a timeline interface for displaying a timeline representing a program and for  
15 associating enhanced content with a position in time on the timeline and for associating time-based media with a position in time on the timeline; and

means for allowing a user to place a component representing enhanced content at a position in time on the timeline, wherein the component representing enhanced content includes a field for storing a reference to a file defining the enhanced content;

20 means for allowing the user to place a component representing time-based media at a position in time on the timeline, wherein the component representing time-based media includes a field for storing a reference to a file for storing the time-based media; and

means for allowing the user to perform editing operations on the timeline that  
25 affect temporal properties of components on the timeline, wherein the editing operations process components representing enhanced content and components representing time-based media in the same manner.

35. The editing system of claim 34, wherein the timeline includes at least one track for  
30 enhanced content and at least one track for time-based media.

36. The editing system of claim 34, wherein the time-based media represented by the component is associated with the position in time in the program and the component representing the time-based media is displayed on the timeline according to the position in time.

5

37. The editing system of claim 34, wherein the enhanced content represented by the component is associated with the position in time in the program and the component is representing the enhanced content is displayed on the timeline according to the position in time.

10

38. The editing system of claim 34, wherein the means for allowing the user to place the component representing enhanced content on the timeline comprises:

means for allowing the user to place the component representing enhanced content at the position in time on the timeline according to a selection of either a point in time or a duration on the timeline, such that the enhanced content represented by the component is associated with the point in time or the duration in the program according to the selection and an indication of the component is displayed on the timeline according to the selection.

15

20 39. The editing system of claim 38, wherein the component representing enhanced content is placed on the timeline at a point in time using a locator component.

40. The editing system of claim 38, wherein the component representing enhanced content is placed on the timeline with a duration using a source clip component.

25

41. The editing system of claim 34, wherein the component representing enhanced content on the timeline includes a reference to a native application capable of processing the enhanced content.

30 42. The editing system of claim 34, wherein the component representing time-based media is placed on the timeline with a duration using a source clip component.

43. The editing system of claim 42, wherein the component representing enhanced content is placed on the timeline with a duration using a source clip component.

5 44. The editing system of claim 34, wherein the component representing time-based media is placed on the timeline at the point in time using a locator component.

45. The editing system of claim 44, wherein the component representing enhanced content is placed on the timeline at a point in time using a locator component.

10

46. The editing system of claim 34, wherein the component representing enhanced content has a type and wherein the editing system includes means for processing a component representing enhanced content according to the type.

15 47. The editing system of claim 46, wherein if the type of the component is a trigger, then the means for processing the component includes means for specifying a display region for video and a display region for enhanced content represented by the component.

48. The editing system of claim 46, wherein if the type of the component is opaque, then  
20 the means for processing the component includes means for identifying a native application for the referenced file and executing the native application to permit access to the referenced file.

49. The editing system of claim 34, wherein the component representing enhanced  
25 content and the component representing time-based media are instances of subclasses of an object class for which the editing operations are implemented.

50. The editing system of claim 34, wherein the component representing enhanced  
content and the component representing time-based media are instances of the same  
30 object class for which the editing operations are implemented.

51. The editing system of claim 34, wherein the component representing enhanced content and the component representing time-based media are implemented using the same type of component for which the editing operations are defined.

5 52. The editing system of claim 34, wherein the editing operations include trim operations.

53. The editing system of claim 34, wherein the editing operations include slip operations.

10

54. The editing system of claim 34, wherein the editing operations include slide operations.

15 55. The editing system of claim 34, wherein the editing operations include cut operations.

56. The editing system of claim 34, wherein the editing operations include lift operations.

20 57. The editing system of claim 34, wherein the editing operations include overwrite operations.

58. The editing system of claim 34, wherein the editing operations include splice operations.

25

59. The editing system of claim 34, wherein the editing operations include auto-synchronization operations.

30 60. A method for editing a time-based program of time-based media and enhanced content, comprising:

displaying a timeline representing a program and for associating enhanced content with a position in time on the timeline and for associating time-based media with a position in time on the timeline; and

allowing a user to place a component representing enhanced content at a position  
5 in time on the timeline, wherein the component representing enhanced content includes a field for storing a reference to a file defining the enhanced content;

allowing the user to place a component representing time-based media at a position in time on the timeline, wherein the component representing time-based media includes a field for storing a reference to a file for storing the time-based media; and

10 allowing the user to perform editing operations on the timeline that affect temporal properties of components on the timeline, wherein the editing operations process components representing enhanced content and components representing time-based media in the same manner.

15 61. A computer program product, comprising:

a computer-readable medium;

computer program instructions stored on the computer readable medium that, when executed by a computer, instruct the computer to perform a method for editing a time-based program of time-based media and enhanced content, comprising:

20 displaying a timeline representing a program and for associating enhanced content with a position in time on the timeline and for associating time-based media with a position in time on the timeline; and

allowing a user to place a component representing enhanced content at a position in time on the timeline, wherein the component representing enhanced content includes a  
25 field for storing a reference to a file defining the enhanced content;

allowing the user to place a component representing time-based media at a position in time on the timeline, wherein the component representing time-based media includes a field for storing a reference to a file for storing the time-based media; and

allowing the user to perform editing operations on the timeline that affect  
30 temporal properties of components on the timeline, wherein the editing operations

process components representing enhanced content and components representing time-based media in the same manner.

62. In an editing system for editing a program of time-based media and enhanced content  
5 and having a bin for storing representations of content accessible by the editing system  
for placement in the program, a method for importing enhanced content into the bin of  
the editing system using an enhancement file, wherein the enhancement file includes data  
defining one or more enhancements, wherein an enhancement comprises attributes  
including an identifier of the enhancement and a reference to a file defining enhanced  
10 content associated with the enhancement, the method comprising:

processing the enhancement file to locate each enhancement; and  
for each enhancement, creating a component in the bin of the editing system,  
wherein the component includes information describing the enhanced content.

15 63. The method of claim 62, wherein the information describing the enhanced content  
comprises a reference to the enhancement file and the identifier of the enhancement.

64. The method of claim 63, wherein the information describing the enhanced content  
comprises the reference to the file defining the enhanced content associated with the  
20 enhancement.

65. The method of claim 64, wherein the information describing the enhanced content  
comprises an identifier of a native application for processing the file defining the  
enhanced content associated with the enhancement.

25

66. The method of claim 65, wherein the information describing the enhanced content  
comprises a type of the enhancement.

67. The method of claim 66, wherein the information describing the enhanced content  
30 comprises custom attributes for the enhancement.

68. The method of claim 62, wherein the information describing the enhanced content comprises the reference to the file defining the enhanced content associated with the enhancement.

5 69. The method of claim 68, wherein the information describing the enhanced content comprises an identifier of a native application for processing the file defining the enhanced content associated with the enhancement.

10 70. The method of claim 69, wherein the information describing the enhanced content comprises a type of the enhancement.

71. The method of claim 70, wherein the information describing the enhanced content comprises custom attributes for the enhancement.

15 72. The method of claim 62, wherein the information describing the enhanced content comprises a type of the enhancement.

73. The method of claim 62, wherein the information describing the enhanced content comprises custom attributes for the enhancement.

20

74. A computer program product, comprising:

a computer-readable medium;

computer program instructions stored on the computer readable medium that, when executed by a computer, define an editing system for editing a program of time-based media and enhanced content and having a bin for storing representations of content accessible by the editing system for placement in the program, and that instructs the computer to perform a method for importing enhanced content into the bin of the editing system using an enhancement file, wherein the enhancement file includes data defining one or more enhancements, wherein an enhancement comprises attributes including an identifier of the enhancement and a reference to a file defining enhanced content associated with the enhancement, the method comprising:

25

30

processing the enhancement file to locate each enhancement; and  
for each enhancement, creating a component in the bin of the editing system,  
wherein the component includes information describing the enhanced content.

5 75. An editing system for editing a time-based program of time-based media and  
enhanced content, comprising:

a bin for storing representations of content accessible by the editing system for  
placement in the program; and

10 means for importing enhanced content into the bin of the editing system using an  
enhancement file, wherein the enhancement file includes data defining one or more  
enhancements, wherein an enhancement comprises attributes including an identifier of  
the enhancement and a reference to a file defining enhanced content associated with the  
enhancement, including:

15 means for processing the enhancement file to locate each enhancement;  
and  
means for creating, for each enhancement, a component in the bin of the  
editing system, wherein the component includes information describing  
the enhanced content.

20 76. A digital information product representing one or more enhancements describing  
enhanced content for placement in a program of time-based media and enhanced content  
using an editing system, comprising:

a computer readable medium;

25 information stored on the computer readable medium that, when accessed by the  
computer is interpreted as an enhancement file, comprising:

data defining one or more enhancements, wherein an enhancement comprises one  
or more attributes, including:

30 an identifier of the enhancement within the enhancement file; and  
a reference to a file defining enhanced content associated with the  
enhancement.

77. The digital information product of claim 76, wherein the one or more attributes further comprises:

an identifier of a native application for processing the file defining the enhanced content.

5

78. The digital information product of claim 77, wherein the one or more attributes further comprises:

a type indicating a type of the enhancement.

10 79. The digital information product of claim 78, wherein the one or more attributes further comprises:

one or more custom attributes for the enhancement.

15 80. The digital information product of claim 77, wherein the one or more attributes further comprises:

one or more custom attributes for the enhancement.

81. The digital information product of claim 76, wherein the one or more attributes further comprises:

20 a type indicating a type of the enhancement.

82. The digital information product of claim 81, wherein the one or more attributes further comprises:

one or more custom attributes for the enhancement.

25

83. The digital information product of claim 76, wherein the one or more attributes further comprises:

one or more custom attributes for the enhancement.

30 84. The digital information product of claim 76, wherein the one or more enhancements comprises a list of enhancements.

85. The digital information product of claim 84, wherein the one or more attributes further comprises:

5 an identifier of a native application for processing the file defining the enhanced content.

86. The digital information product of claim 85, wherein the one or more attributes further comprises:

10 a type indicating a type of the enhancement.

87. The digital information product of claim 86, wherein the one or more attributes further comprises:

one or more custom attributes for the enhancement.

15 88. The digital information product of claim 85, wherein the one or more attributes further comprises:

one or more custom attributes for the enhancement.

20 89. The digital information product of claim 84, wherein the one or more attributes further comprises:

a type indicating a type of the enhancement.

90. The digital information product of claim 89, wherein the one or more attributes further comprises:

25 one or more custom attributes for the enhancement.

91. The digital information product of claim 84, wherein the one or more attributes further comprises:

one or more custom attributes for the enhancement.

30

92. In an editing system for editing a program of time-based media and enhanced content having at least one track for enhanced content and at least one track for time-based media, a method for generating a document in a markup language describing the at least one track for enhanced content, the method comprising:

5 identifying each component representing enhanced content on the at least one track for enhanced content;

for each identified component representing enhanced content, creating an element in the markup language document comprising:

10 a start position of the component on the at least one track for enhanced content;

if the component is a source clip, then an end position of the component on the at least one track for enhanced content; and

15 a reference to data in the markup language document defining an enhancement that describes the enhanced content represented by the component.

93. The method of claim 92, wherein the start position is specified in a plurality of time code formats.

20 94. The method of claim 93, wherein the end position is specified in a plurality of time code formats.

95. The method of claim 94, wherein the plurality of time code formats includes 24 frames per second, 25 frames per second and 30 frames per second.

25 96. The method of claim 92, wherein the data defining the enhancement comprises: one or more attributes, including:

an identifier of the enhancement within the enhancement file; and

a reference to a file defining the enhanced content associated with the enhancement.

30

97. The method of claim 96, wherein the one or more attributes further includes:

an identifier of a native application for processing the file defining the enhanced content.

98. The method of claim 97, wherein the one or more attributes further includes:

5 a type indicating a type of the enhancement.

99. The method of claim 98, wherein the one or more attributes further includes:

one or more custom attributes for the enhancement.

10 100. The method of claim 97, wherein the one or more attributes further includes:

one or more custom attributes for the enhancement.

101. The method of claim 96, wherein the one or more attributes further includes:

a type indicating a type of the enhancement.

15

102. The method of claim 101, wherein the one or more attributes further includes:

one or more custom attributes for the enhancement.

103. The method of claim 96, wherein the one or more attributes further includes:

20 one or more custom attributes for the enhancement.

104. The method of claim 92, further comprising:

identifying each of the at least one tracks for enhanced content;

for each identified track for enhanced content, creating an interactive timeline

25 element in the markup language document comprising:

a reference to each element in the markup language document that describes a component representing enhanced content on the identified track for enhanced content.

30 105. A computer program product, comprising:

a computer readable medium;

computer program instructions stored on the computer readable medium that, when executed by a computer, define an editing system for editing a program of time-based media and enhanced content having at least one track for enhanced content and at least one track for time-based media, and instructs the computer to perform a method for  
5 generating a document in a markup language describing the at least one track for enhanced content, the method comprising:

identifying each component representing enhanced content on the at least one track for enhanced content;

for each identified component representing enhanced content, creating an element  
10 in the markup language document comprising:

a start position of the component on the at least one track for enhanced content;

if the component is a source clip, then an end position of the component on the at least one track for enhanced content; and

15 a reference to data in the markup language document defining an enhancement that describes the enhanced content represented by the component.

106. An editing system for editing a time-based program of time-based media and enhanced content, comprising:

20 a timeline interface having at least one track for enhanced content and at least one track for time-based media; and

means for generating a document in a markup language describing the at least one track for enhanced content, including:

25 means for identifying each component representing enhanced content on the at least one track for enhanced content; and

means for creating, for each identified component representing enhanced content, an element in the markup language document comprising:

a start position of the component on the at least one track for enhanced content;

30 if the component is a source clip, then an end position of the component on the at least one track for enhanced content; and

a reference to data in the markup language document defining an enhancement that describes the enhanced content represented by the component.

5 107. An editing system, comprising:

a timeline interface for displaying a timeline representing a program and for associating enhanced content with a position in time on the timeline and for associating time-based media with a position in time on the timeline; and

10 a bin for storing representations of enhanced content and time-based media accessible by the editing system for placement in the program;

means for importing enhanced content into the bin to create one or more components in the bin, wherein a component in the bin includes information describing the enhanced content including an identifier associated with the enhanced content;

15 means for allowing a user to place a component representing enhanced content at a position in time on the timeline, such that the enhanced content represented by the component is associated with the position in time in the program and an indication of the component is displayed on the timeline according to the position in time; and

means for updating the information describing the enhanced content related to a component in the bin using the identifier associated with the enhanced content.

20

108. The editing system of claim 107, wherein the timeline includes at least one track for enhanced content and at least one track for time-based media.

25 109. The editing system of claim 107, wherein the identifier includes a reference to an enhancement file including data defining one or more enhancements, wherein an enhancement describes the enhanced content, and wherein the identifier includes an identifier of the enhancement in the enhancement file.

30 110. The editing system of claim 109, wherein the information describing the enhanced content comprises a reference to a file defining the enhanced content.

111. The editing system of claim 110, wherein the information describing the enhanced content comprises an identifier of a native application for processing the file defining the enhanced content.

5 112. The editing system of claim 111, wherein the information describing the enhanced content comprises a type.

113. The editing system of claim 112, wherein the information describing the enhanced content comprises custom attributes.

10

114. The editing system of claim 107, wherein the information describing the enhanced content comprises a reference to a file defining the enhanced content.

115. The editing system of claim 114, wherein the information describing the enhanced  
15 content comprises an identifier of a native application for processing the file defining the enhanced content.

116. The editing system of claim 115, wherein the information describing the enhanced content comprises a type.

20

117. The editing system of claim 116, wherein the information describing the enhanced content comprises custom attributes.

118. The editing system of claim 107, wherein the information describing the enhanced  
25 content comprises a type.

119. The editing system of claim 118, wherein the information describing the enhanced content comprises custom attributes.

30 120. A method for editing a time-based program of time-based media and enhanced content, comprising:

displaying a timeline representing a program and for associating enhanced content with a position in time on the timeline and for associating time-based media with a position in time on the timeline; and

storing representations of enhanced content and time-based media in a bin  
5 accessible by the editing system for placement in the program;

importing enhanced content into the bin to create one or more components in the bin, wherein a component in the bin includes information describing the enhanced content including an identifier associated with the enhanced content;

allowing a user to place a component representing enhanced content at a position  
10 in time on the timeline, such that the enhanced content represented by the component is associated with the position in time in the program and an indication of the component is displayed on the timeline according to the position in time; and

updating the information describing the enhanced content related to a component in the bin using the identifier associated with the enhanced content.

15

121. A computer program product, comprising:

a computer-readable medium;

computer program instructions stored on the computer readable medium that, when executed by a computer, instruct the computer to perform a method for editing a  
20 time-based program of time-based media and enhanced content, comprising:

displaying a timeline representing a program and for associating enhanced content with a position in time on the timeline and for associating time-based media with a position in time on the timeline; and

storing representations of enhanced content and time-based media in a bin  
25 accessible by the editing system for placement in the program;

importing enhanced content into the bin to create one or more components in the bin, wherein a component in the bin includes information describing the enhanced content including an identifier associated with the enhanced content;

allowing a user to place a component representing enhanced content at a position  
30 in time on the timeline, such that the enhanced content represented by the component is

associated with the position in time in the program and an indication of the component is displayed on the timeline according to the position in time; and

updating the information describing the enhanced content related to a component in the bin using the identifier associated with the enhanced content.

5

122. An editing system comprising:

a timeline interface for displaying a timeline representing a program and for associating enhanced content with a position in time on the timeline and for associating time-based media with a position in time on the timeline;

10 means for allowing a user to relate enhanced content and time-based media to create auto-synchronized content;

means for allowing a user to place a component representing auto-synchronized content at a position in time on the timeline such that the auto-synchronized content represented by the component is associated with the position in time in the program; and

15 means for allowing a user to perform editing operations on the time-based media of the auto-synchronized content on the timeline such that the editing operations are also performed on the enhanced content of the auto-synchronized content.

123. The editing system of claim 122, wherein the timeline includes at least one track for enhanced content and at least one track for time-based media.

124. The editing system of claim 123, wherein the enhanced content of the auto-synchronized content is placed on the at least one enhanced content track and the time-based media of the auto-synchronized content is placed on the at least one time-based media track

125. The editing system of claim 122, wherein enhanced content comprises nontemporal media data.

30 126. The editing system of claim 122, wherein enhanced content comprises data that is not processed natively by the editing system.

127. The editing system of claim 122, wherein enhanced content comprises interactive content having a time during the program that results in content being perceived during playback of the program.

5

128. The editing system of claim 122, wherein enhanced content comprises data that is opaque to the editing system.

129. The editing system of claim 128, wherein the opaque enhanced content uses an application other than the editing system for viewing the enhanced content.

10

130. The editing system of claim 128, wherein the opaque enhanced content uses an application other than the editing system for editing the enhanced content.

131. The editing system of claim 128, wherein the opaque enhanced content is not perceived as part of the program by the user during playback of the program in the editing system.

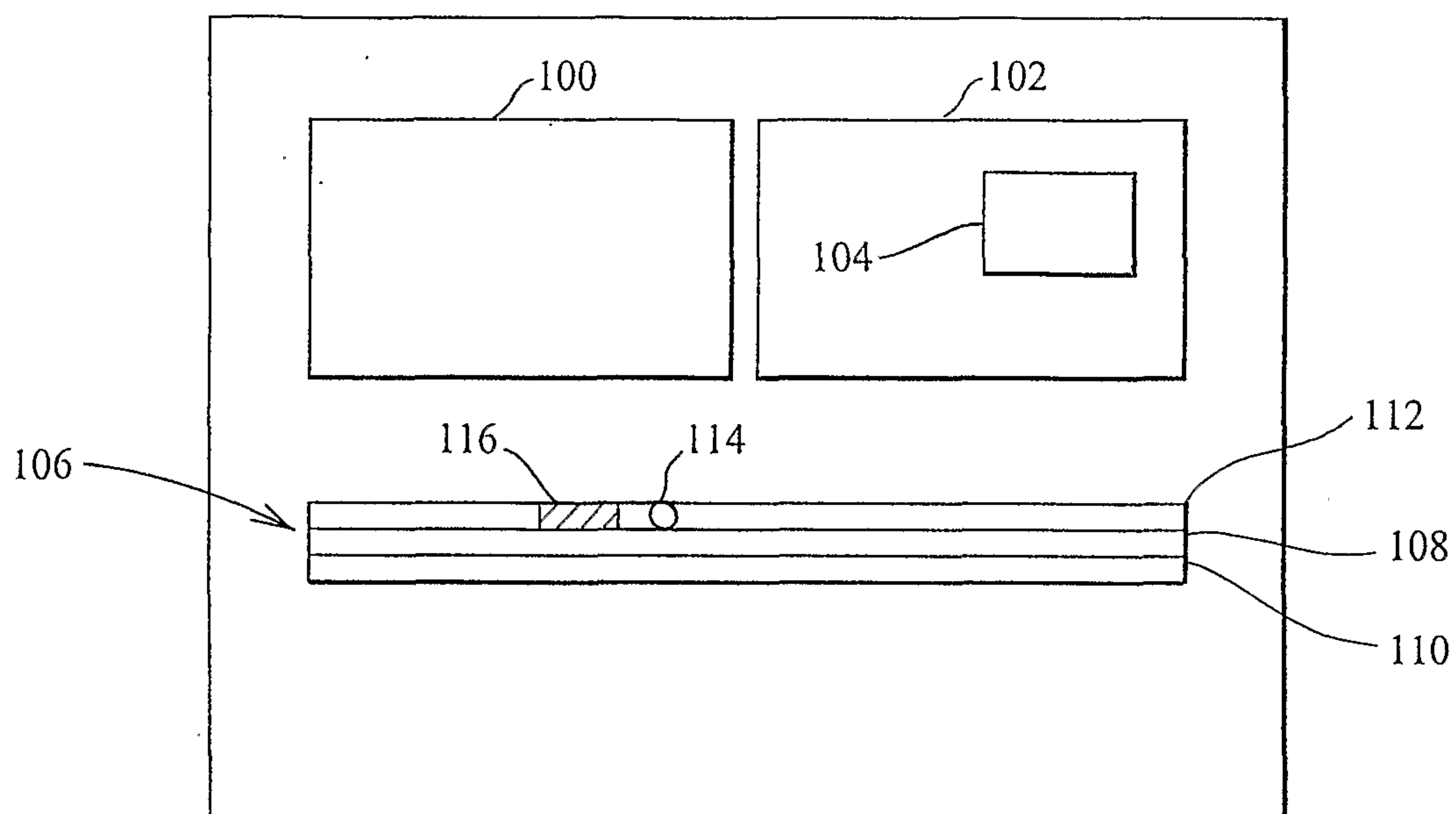
15

132. The editing system of claim 128, wherein the opaque enhanced content supplies information to create the program.

20

133. The editing system of claim 128, wherein the opaque enhanced content tracks information about the program.

25

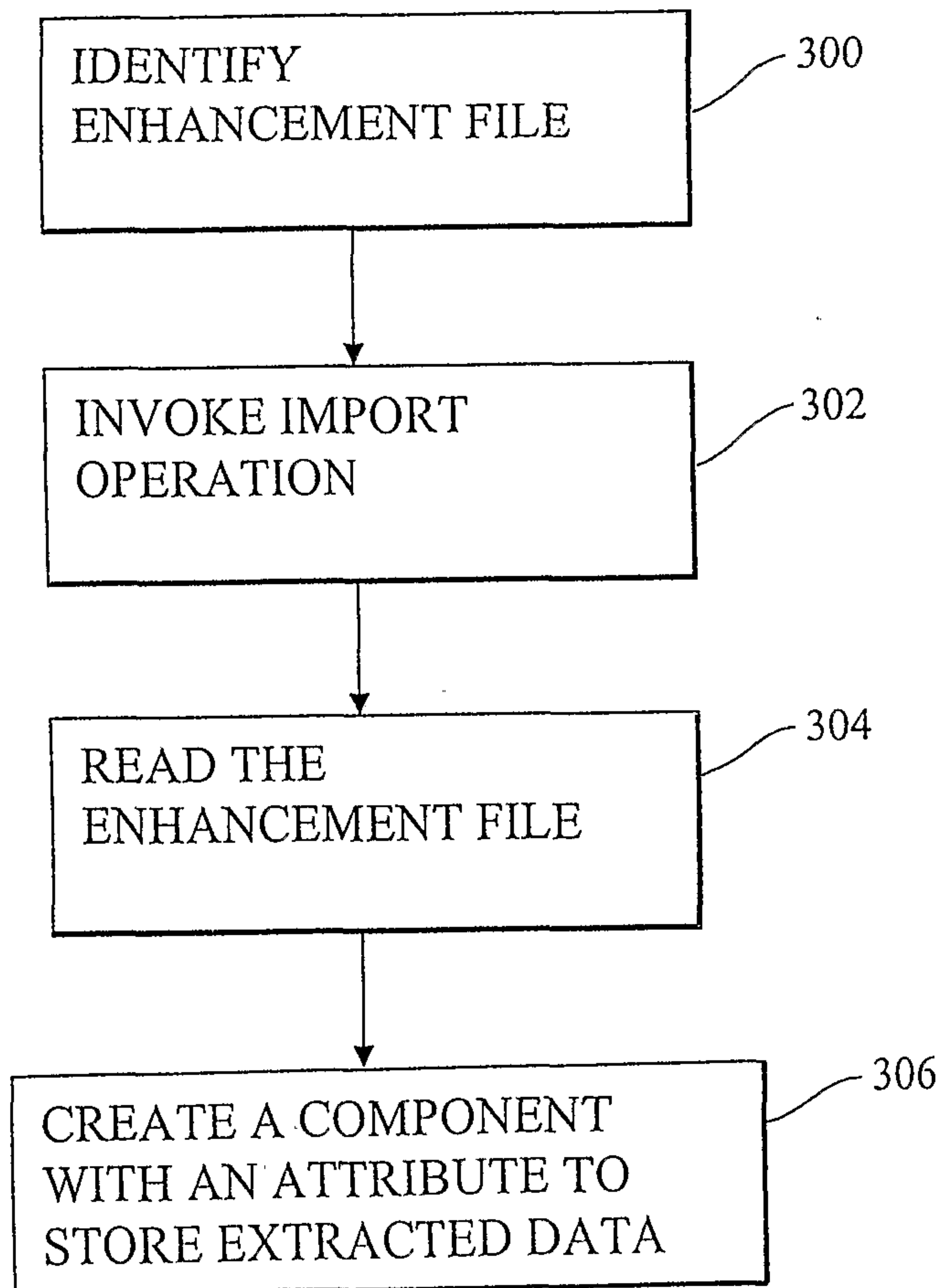
**FIG. 1**

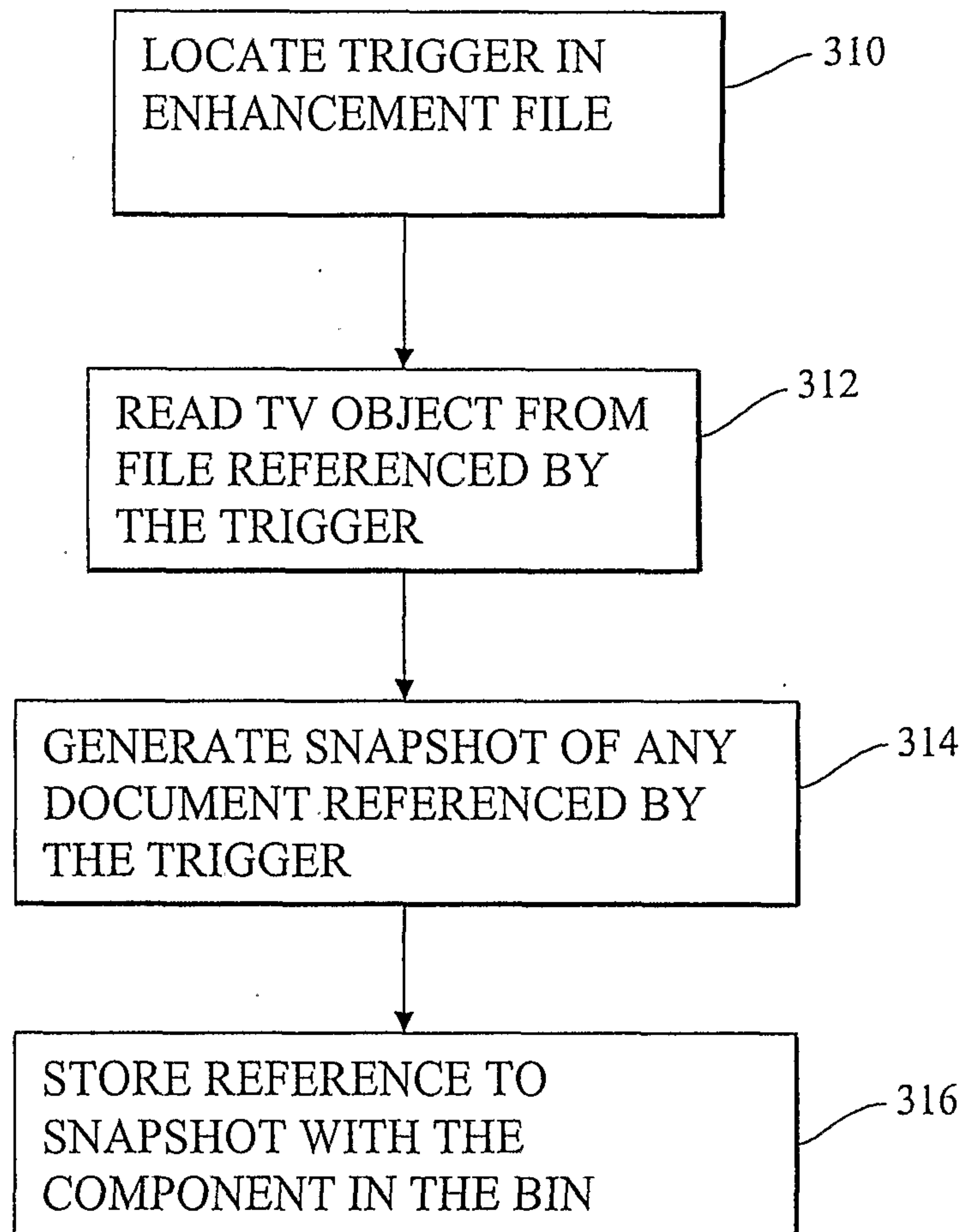
|     |                 |
|-----|-----------------|
| 200 | TYPE            |
| 201 | LOCATOR         |
| 202 | APPLICATION     |
| 204 | NAME            |
| 206 | EXPIRATION      |
| 208 | EXPIRATION TYPE |
| 210 | IDENTIFIER      |
| 212 | USER DATA       |
| 214 | CHECKSUM        |

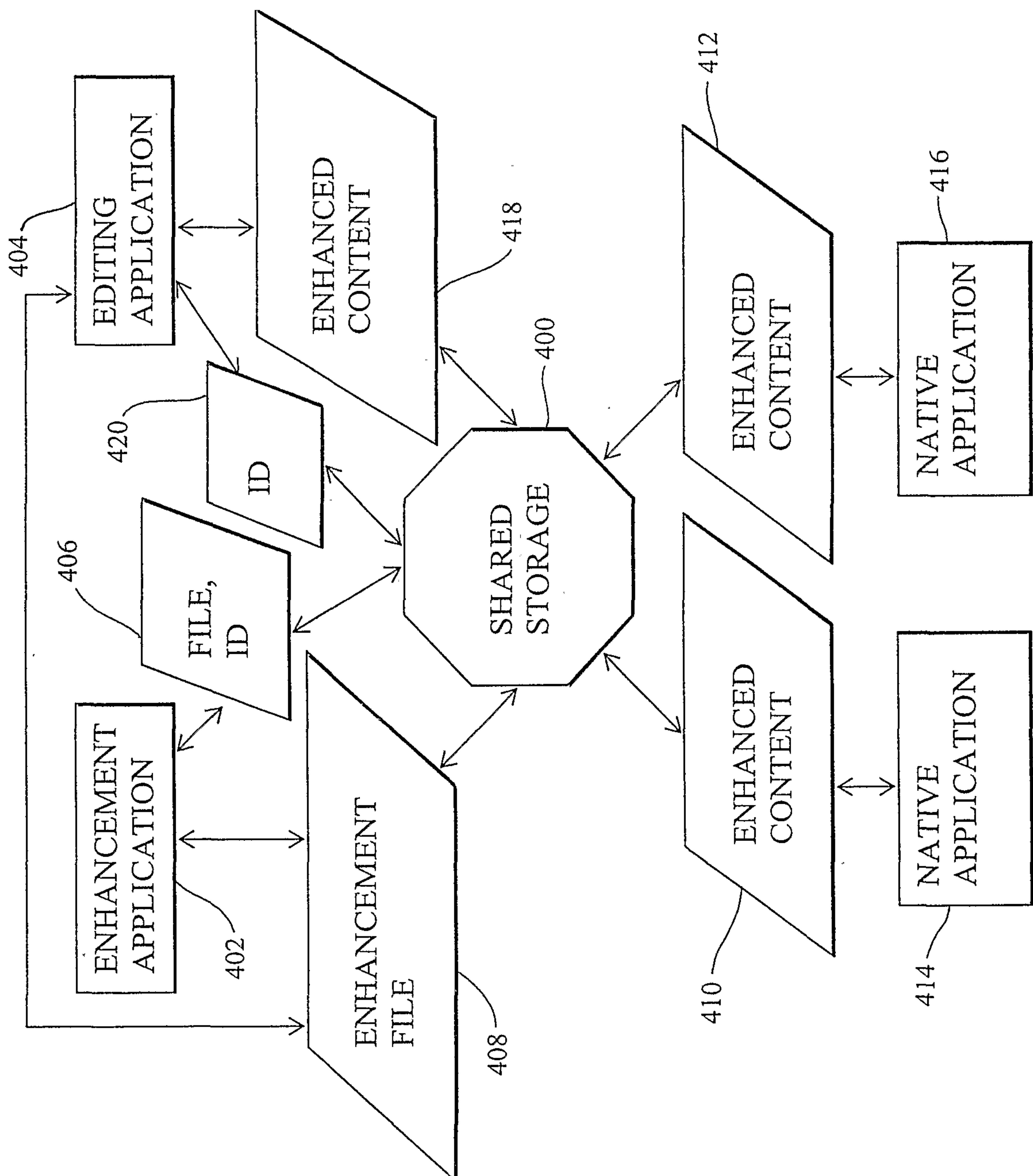
**FIG. 2A**

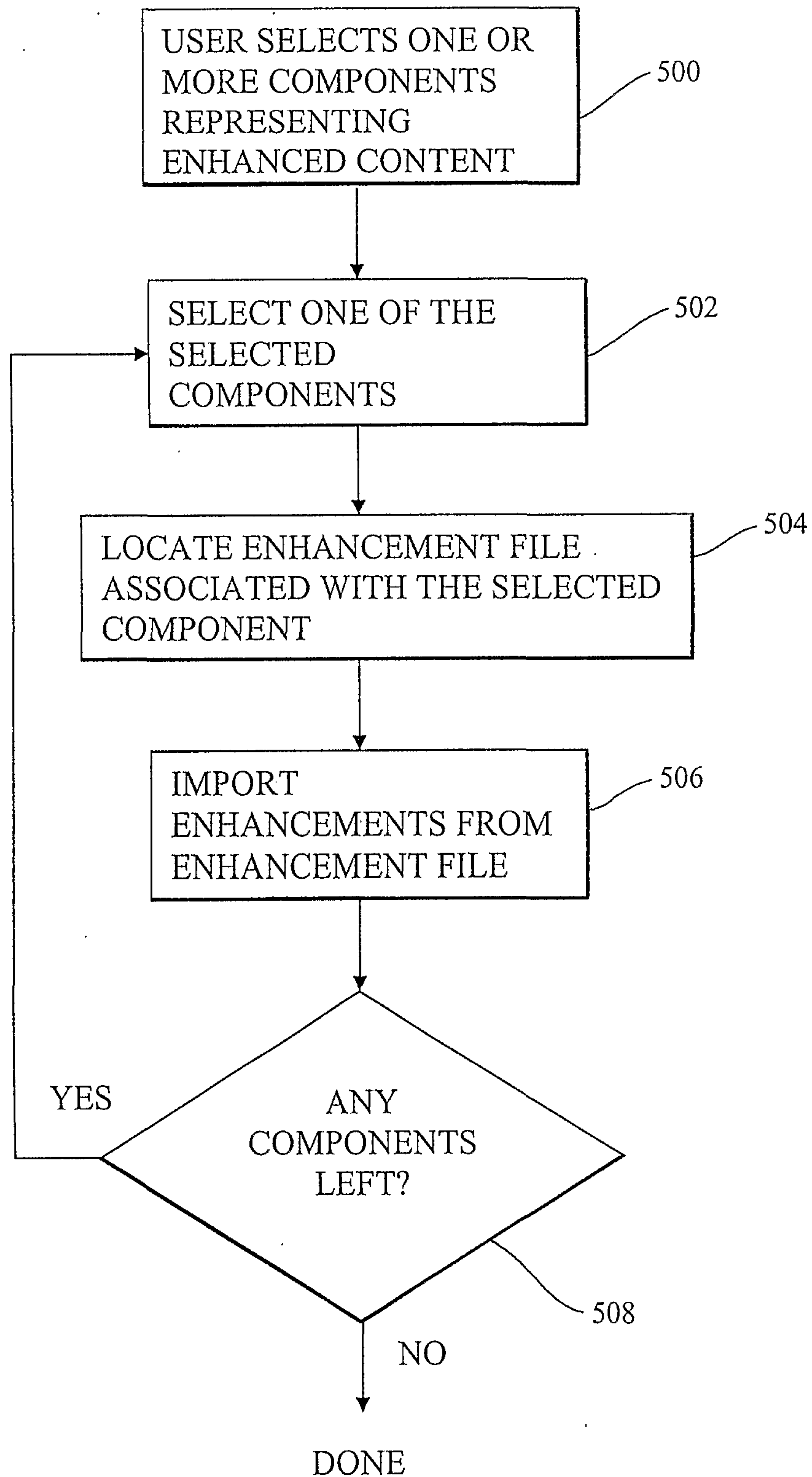
|     |                 |
|-----|-----------------|
| 220 | SNAPSHOT        |
| 222 | RELATED FILES   |
| 224 | CROPPING        |
| 228 | OVERLAY         |
| 226 | SCALING         |
| 232 | PAGE DIMENSIONS |
| 230 | TV DIMENSIONS   |
| 234 | LAYOUT          |

**FIG. 2B**

**FIG. 3A**

**FIG. 3B**

**FIG. 4**

**FIG. 5**

