METHODS, APPARATUS, AND SYSTEMS FOR STORING, RETRIEVING AND PLAYING MULTIMEDIA DATA

BACKGROUND OF THE INVENTION

Field of the Invention

The invention relates generally to improvements in computer systems. More particularly, the invention relates to methods, apparatus, and systems storing multimedia content such as audio, text, image, and graphical content in a cache directory.

Discussion of the Related Art

Prior art graphics processing storage medium, sometimes called a cache system, is known to those skilled in the art. For example, a conventional caching system is typically composed of a small fast storage device that contains a "snapshot" of information originally received from a larger, slower source. The snapshot is considered by the particular implementation to be the most relevant information to the processing occurring during the current time period.

In the context of Internet content, a "cache" is a file, database, directory, or set of directories disposed in a computer file system. The cache stores content that has been previously retrieved, generated, or otherwise produced. Internet browsers and editors use cache directories to store content. The cached content is used in place of remote content whenever possible in order to decrease retrieval latencies. Therefore, many web browsers and text editors save Internet and other text and graphical content in a cache directory in order to reduce access times. This content is usually stored in its original form, for example, hypertext markup language (HTML) and accompanying images.

A problem with this technology has been that to view content based on the image data stored in the cache typically requires layout and rendering of the

data. If the data upon which content is based does not change, the process of rendering need only occur once to a display buffer. When information is changed, the information must be re-rendered to reflect the desired change. For complex graphics scenes re-rendering can require massive processing for only incremental changes in the scene or particular graphic. The layout and rendering processes are time consuming and require processor resources. Therefore, what is required is a solution that provides play of multimedia content more efficiently in terms of time and processor resources.

Heretofore, the requirements of timely and processor efficient play of multimedia content have not been fully met. What is needed is a solution that simultaneously addresses these requirements. The invention is directed to meeting these requirements, among others.

SUMMARY OF THE INVENTION

A primary goal of the invention is to provide timely and processor efficient display of multimedia content. In accordance with these goals, there is a particular need for a storage medium that includes rendered multimedia content and the semantic content of the multimedia content. A storage medium including both the multimedia content and the semantic content is referred to herein as a rendered cache.

For various embodiments of the invention, the semantic content can include locations, sizes, shapes, and target universal resource identifiers of hyperlinks, multimedia element timing, and other content play instructions. The very fast play of content stored in the rendered cache is due to the elimination of the steps of laying out the content, rendering the content, and generating the semantic representation of the content. These steps are required each time the content is played after retrieval from a conventional cache. The only steps required for playing content from the rendered cache are to read the rendered content, read the semantic content, restore the semantic representation, and play the content.

A web browser visiting a web page that resides in a rendered cache provides an almost instantaneous display of the web page. The caching

mechanism provided by various embodiments of the invention is independent of content file format and the stored semantic content file format. As long as a client application, such as a content browser, can recognize and play the multimedia content and recognize and interpret the semantic content, the application can realize the benefits provided by the rendered cache. Thus, it is possible to simultaneously satisfy the above-discussed requirements of timely and processor efficient display of multimedia content, which, in the case of the prior art, are not simultaneously satisfied.

A first aspect of the invention is provided as an embodiment that is based on a method, implemented in at least one computer, for storing multimedia data. The method for storing multimedia data comprises detecting multimedia content, generating a semantic representation of a rendered representation of the multimedia content from the play instructions, storing the rendered representation in a storage medium, and storing data corresponding to the semantic representation in the storage medium. The multimedia content includes play instructions and at least one multimedia element. The at least one multimedia element includes at least one of graphical images, audio, text, and full motion video. The play instructions include at least one of timing of the multimedia content and ordering of the multimedia content. The semantic representation describes at least one of characteristics of the rendered representation, and relationships between different multimedia elements disposed in the rendered representation.

A second aspect of the invention is provided as an embodiment that is based on a method, implemented in at least one computer, for storing multimedia data. The method for storing multimedia data comprises detecting multimedia content including layout instructions, and laying out the multimedia content according to the layout instructions to form rendering instructions and a semantic representation of a rendered representation of the multimedia content. The method also includes rendering the multimedia content according to the rendering instructions to produce the rendered representation, storing the rendered representation in a storage medium, and storing data corresponding to the semantic representation in the storage medium.

A third aspect of the invention is provided as an embodiment that is based on a method, implemented in at least one computer, for retrieving multimedia data. The method for retrieving multimedia data comprises processing resources of a first computer of the at least one computer detecting a request for requested multimedia content, and processing resources coupled with the first computer determining whether data corresponding to the requested multimedia content is disposed in a storage medium. The storage medium is coupled with the first computer and includes rendered representations of multimedia content and semantic content. Embodiments according to the third aspect of the invention also include responding to a determination that data corresponding to the requested multimedia content are disposed in the storage medium by retrieving a rendered representation of the requested multimedia content; and retrieving semantic content corresponding to the requested multimedia content.

A fourth aspect of the invention is implemented in an embodiment that is based on a rendered cache comprising a storage medium, and an indexing mechanism adapted to store and retrieve a rendered representation of the multimedia content formatted for rapid play and semantic content of the multimedia content.

A fifth aspect of the invention is implemented in an embodiment that is based on a client. The client comprises processing resources adapted to detect a rendered representation of multimedia content and semantic content of the rendered representations, and processing resources adapted to respond to detecting the rendered representation of the multimedia content and the semantic content by playing at least a portion of the rendered representation according to the semantic content.

A sixth aspect of the invention is implemented in an embodiment that is based on a system for using multimedia content. The system comprises web crawler processing resources adapted to access the multimedia content from source data storage, rendering processing resources, and a rendered cache as described above as the fourth aspect of the invention. The rendering processing resources are adapted to generate a semantic representation of a rendered

representation of the multimedia content, and format the semantic representation as semantic content, and render the multimedia content into the rendered representation, the rendered representation is formatted for rapid play.

A seventh aspect of the invention is implemented in an embodiment that is based on a system for accessing multimedia content. The system for accessing multimedia comprises a rendered cache as described above as the fourth aspect of the invention, and rendering processing resources adapted to convert the multimedia content into the rendered representation, the rendered representation is formatted for rapid play, and create a graphical representation of the multimedia content.

An eighth aspect of the invention is implemented in a method for playing multimedia content. The method comprises retrieving a rendered representation of the multimedia content from a storage medium, and retrieving semantic content of the rendered representation from the storage medium. The method includes browser processing resources reading the rendered representation and the semantic content, and the browser processing resources restoring a semantic representation based on the semantic content. The method includes the browser processing resources transmitting an active portion of the rendered representation to a client, and transmitting an active portion of the semantic content corresponding to the active portion of the rendered representation to the client. The active portion of the rendered representation is one of a portion of the rendered representation presently being played, and a portion of the rendered representation to be played rapidly after transmitting. The method also includes client processing resources detecting the active portion of the rendered representation and the active portion of the semantic content, and the client processing resources playing the active portion of the rendered representation.

BRIEF DESCRIPTION OF THE DRAWINGS

These, and other, goals and aspects of the invention will be better appreciated and understood when considered in conjunction with the following description and the accompanying drawings. Various embodiments of the

invention are illustrated in the drawings accompanying and forming a part of this specification, wherein like reference characters (if they occur in more than one view) designate the same parts. It should be noted that the features illustrated in the drawings are not necessarily drawn to scale.

FIG. 1 illustrates a schematic block diagram of a conventional method for retrieving and playing multimedia content, appropriately labeled "PRIOR ART".

FIG. 2 illustrates a schematic block diagram of a process overview for retrieving and playing multimedia content using a rendered cache, representing an embodiment of the invention.

FIG. 2A illustrates a schematic block diagram of a process overview of a paint stream process, according to an embodiment of the invention.

FIG. 3 illustrates a schematic block diagram including render process details, representing an embodiment of the invention.

FIG. 4 illustrates a schematic block diagram including play process details, representing an embodiment of the invention.

FIGS. 5A-5B illustrate screen shots of portions of a Toronto Exchange Internet page, representing an embodiment of the invention.

FIG. 6A illustrates the timing of play of different multimedia elements for an example of multimedia content that does not require layout, representing an embodiment of the invention.

FIGS. 6B-6D illustrate different images included in the multimedia content not including layout example representing an embodiment of the invention.

FIG. 7 illustrates a communications system including a rendered cache, representing an embodiment of the invention.

FIG. 7A illustrates a communications system including a rendered cache, showing a client, representing an embodiment of the invention.

FIG. 7B illustrates a communications system, representing an embodiment of the invention.

FIG.8A illustrates a communications system with render engine located at the set top box, representing an embodiment of the invention.

FIG.8B illustrates a communications system with render engine located at the cable company headend, according to an embodiment of the invention.

Fig. 8C illustrates the location and connection of various components involved in the rendering process, including a partial render engine, according to an embodiment of the invention.

DESCRIPTION OF PREFERRED EMBODIMENTS

The invention and the various features and advantageous details thereof are explained more fully with reference to the non-limiting embodiments that are illustrated in the accompanying drawings and detailed in the following description of preferred embodiments. Descriptions of well-known components and processing techniques are omitted so as not to unnecessarily obscure the invention in detail. It should be understood, however, that the following description, while indicating preferred embodiments of the invention and numerous specific details thereof, is given by way of illustration and not of limitation. Many changes and modifications may be made within the scope of the invention without departing from the spirit thereof, and the invention includes all such modifications.

Methods, apparatus and systems are described for storing multimedia content using a process for caching fully rendered documents in a way that significantly increases content viewing speeds, navigation in a hyperlink document, while decreasing processing requirements.

Definitions

The following terms are used in the description of various embodiments of the invention provided herein.

Content: Text and graphical information that require a layout and/or rendering process in order to be viewed on a computer, television or other display device. Other terms for content include web-page, document, Internet content, hypertext markup language (HTML), eXtensible Markup Language

(XML), and Television Markup Language (TVML). Content can also include non-graphical information such as audio.

Content Browser: A computer program designed to retrieve, display or navigate content. Examples include Internet web browsers, HTML/XML/Standard Generalized Markup Language (SGML) editors, word processors, and Internet web proxies.

HTML: The de facto Internet content standard. HTML includes a set of markup rules that describe the layout of Internet content. Browsers use this markup to layout and render the HTML for viewing computer monitors, televisions, or other displays.

Markup: Notation used to describe the syntactic and semantic features of a content document.

Multimedia Content: Multimedia elements used for playing a presentation for a user. The multimedia elements can include graphical images (including rendered HTML), audio, text, and full motion video.

Navigation: The process of selecting an indexing indication, such as a URI in the form of a hyperlink, from displayed content to access further content.

Paint Stream: A set of rendering instructions that can be used to render multimedia content. These rendering instructions are typically the result of laying out multimedia content (e.g. HTML). The paint stream can also contain semantic information such as the size, position, shape, and target of URIs; size, position, and timing of animated gifs; information about other interactive elements (e.g. HTML forms).

Presentation: Content that references at least one multimedia element. Presentations include play instructions that can be used to define the timing, order, and position of the multimedia plays. The play instructions can include the size, shape and target of all hyperlinks, information on interactive elements (like HTML forms), and Meta values.

Render: The process of generating a graphical representation of data that can be viewed on a display. For example, web browsers render HTML pages into graphical images that can be viewed on a computer monitor or

television. Also the process of generating or converting multimedia data (images, audio, text, full motion video) into a format that can be played.

Rendered Cache: Various embodiments of the invention use the concept of a rendered cache to mean a cache of content that is not only generated (or retrieved) from a multimedia content data source, such as the Internet; but also is rendered and ready for rapid play. The rendered cache can include two types of objects: multimedia content and semantic content. The multimedia content stored in the rendered cache is content that has been rendered and is ready for very quick display. Semantic content includes a description of the semantic features or representation of the rendered content. Examples of semantic features include the location, size, shape and target of hyperlinks, the timing, location, and size of animated graphics interchange format (GIF) frames, the size and relative location of HTML frames, information on HTML forms, HTML meta values, presentation play timing, and other play instructions. A more detailed description of the rendered cache is provided in the Process Description section below.

Semantic Representation: A description of the characteristics, attributes, logical structure, and features of multimedia elements (or objects) that form a rendered representation of multimedia content, or a portion thereof. The data can also describe the relationships between different multimedia elements within a particular presentation portion, and the way various elements of the multimedia content are accessed and manipulated. The semantic representation is typically generated during the layout process and is structured such that the semantic representation can be saved as formatted and indexed semantic content in a file or database, and rapidly restored from the semantic content. The semantic content can be stored along with the multimedia content or as one or more separate indexed files. The semantic representation is independent of the format of the stored semantic content. The Document Object Model (DOM) is one type of semantic representation and is adapted for use with HTML and XML documents.

TVML: Some embodiments of the invention (including the VirtualModem™ presentation system provided by Interactive Channel, Inc.

located in London, Ontario, Canada) use an XML language called television markup language (TVML) to describe multimedia content. TVML includes markup to describe how to play multimedia content. The multimedia content can include text (including HTML), graphical images, audio, text, and full-motion video. TVML can include markup to describe when each multimedia component should be played relative to the other multimedia components.

URI: A Universal Resource Identifier (or URI) is an Internet standard term for all types of names and addresses that refer to content. The term URI encompasses terms such as filename, hyperlink, and Universal Resource Locator (URL).

VMML: An XML markup language (called VMML - VirtualModem™ Markup Language) used to store semantic representations of rendered multimedia content by various embodiments of the invention, such as the VirtualModem™ presentation system.

XML: A markup language used to describe other markup languages, such as HTML and TVML.

Process Description

Various embodiments of the invention include methods, implemented in at least one computer, for storing and retrieving multimedia data. These methods navigate and play multimedia content with increased speed and decreased computer processing by using different types of data objects to represent the multimedia data. A first data object type includes pre-rendered multimedia content data. A second data object type includes a semantic representation of the pre-rendered multimedia content. These data object types can be stored as separate files or can be contained in the same file.

Prior art methods for retrieving and playing multimedia content are represented by Figure 1, which includes a traditional cache 110. After detecting a request to play multimedia content (at step 120), retrieving processing resources, such as those disposed in a web browser, retrieve the corresponding multimedia content data. A traditional web browser, such as Netscape Navigator, Netscape Communicator, or Microsoft® Internet Explorer, when

coupled with a traditional cache 110, then performs the steps described below in response to each and every play request 120.

After retrieving the content (e.g., the HTML content description), the content is read (step 130) from either a traditional cache 110, an Internet 105, or another content data source. Processing resources disposed in a computer can layout the content (step 140), e.g., according to the content's HTML description. During the layout 140, the processing resources generate rendering instructions 140A and derive a semantic representation 140B of the multimedia content. Note that for some embodiments, layout 140 is not required. For these embodiments, the semantic representation 140B can be generated from play instructions, as shown in Figure 3 (at step 315).

Content browsers can use the semantic representation 140B to determine location, size, shape and targets of hyperlinks; and content play instructions. The semantic representation 140B can also be used to describe other interactive presentation elements, e.g., HTML forms. The semantic features corresponding to the depicted graphical representation generated for play when using traditional content browsers coupled with traditional caches 110 persist only as long as the content is being viewed. Because the semantic features must be present whenever the multimedia content is played, and because traditional caches 110 store the multimedia content in a non-rendered original form, traditional browsers must re-render the graphical representations each time a user requests the content, as shown in Figure 1.

A render engine then renders the multimedia content (at step 150) according to the rendering instructions 140A to form rendered content 160 (otherwise referred to herein as the rendered representation of the multimedia content). Finally, a multimedia play engine uses both the rendered content 160 and the semantic representation 140B to play the rendered content (at step 170). For multimedia content including images, the playing 170 includes displaying the rendered image on a user screen according to the semantic representation 140B. The rendered content 160 is also referred to herein as a rendered representation of the multimedia content.

In prior art implementations, playing 170 occurs after layout 140 (or other process in which the semantic representation 140B is generated), and rendering 150 have been completed. Rendering 150 and generation of the semantic representation 140B require a relatively long time between when the play request 120 is received and when multimedia content is played 170 compared to the time required to play content using various method embodiments of the invention. Also, greater data processing is required for rendering 150 and generation of the semantic representation 140B for the multimedia content than the processing required for play 170 using various method embodiments of the invention.

The details of retrieving (step 260), rendering 150, and playing 170 multimedia content for some embodiments of the invention are illustrated in Figs 2 through 4 and described below. Fig. 2 provides an overview of the retrieving 260, and the playing 170 processes for multimedia content, e.g., HTML content, using a rendered cache 201. Methods for using multimedia data according to various embodiments of the invention can be implemented in at least one computer having one or more programs for retrieving and playing multimedia content. The benefits of using the rendered cache 201 for subsequent access to the same multimedia content are also described below.

The rendered cache 201 includes not only rendered content 160 (which can include image data) but also some means of reconstructing the semantic representation 140B of the multimedia data. The reconstruction of the semantic representation 140B can be done using proprietary image formats or separate files that describe the semantic features. This semantic representation 140B can include locations, sizes, and destinations of hyperlinks, descriptions of animations or other dynamic content, and other "meta" information. Meta information can include tagging, refresh (client pull replacement) and platform for Internet content selection (PICS) association labels.

Some embodiments of the invention (including VirtualModem™ interactive presentation systems provided by Interactive Channel Technologies, Inc. located in London, Ontario, Canada) use an XML language called VMML to store the semantic content. The VMML semantic content can include

markup to represent the following semantic features of the rendered content 160:

1. Location, size, shape, and target indices (such as URI) of hyperlinks,
2. Size and relative location of HTML frames in the rendered image,
3. Size, location, and timing of animated GIFs,
4. Size, location, and type of HTML form elements,
5. Timing of multimedia content elements, and
6. Other play 170 instructions.

Proper use of content from a rendered cache 201 eliminates the steps of generating a semantic representation 140B, layout 140 (when needed), and rendering 150. On the other hand traditional web browsers using traditional caching mechanisms must perform these steps before playing 170 the content. Eliminating these steps reduces the time and use of processing resources required for playing 170 the multimedia content.

A rendered cache 201 can include of two types of data objects: multimedia content and semantic content. The content can be stored 320 in any format (i.e., the caching mechanism is independent of file format). Typically, the layout 140 and/or rendering 150 processing resources format the semantic presentation 140B for storage in the rendered cache 201 as semantic content. Alternatively, the layout 140 and/or rendering 150 processing resources can transfer the semantic representation 140B to rendered cache 201 server processing resources which then format the semantic presentation into semantic content to be stored in properly indexed files for retrieval 260. For some embodiments of the invention, content browsers (and/or other client applications using content from the rendered cache 201) can include processing resources, such as a program, for detecting the format of the rendered content 160 and for viewing multimedia content.

When a request for content is received (step 210) the content browser can determine (step 220) whether a rendered representation of the content already exists in the rendered cache 201. The browser can also determine (step

230) whether the content in the rendered cache 201 is outdated. The content request received at step 210 can be provided to the browser using a file target index, such as a Universal Resource Identifier (URI).

Once it has been determined whether an updated rendered representation of the requested data already exists in the rendered cache 201, a first and simpler processing path indicated in Figure 2 can be followed. Because the rendered cache 201 contains valid rendered content corresponding to the request, browser engine processing resources can simply read the semantic content and the rendered content 160, restore the semantic representation (step 240), and then play 170 the rendered content corresponding to the requested content.

The process proceeds along a second path if the server-based system (or other processing resources coupled with the rendered cache 201) determines (at step 220) that the requested content is not in the rendered cache 201, or determines (at step 230) that the content stored in the rendered cache is outdated. If the requested content is not in the rendered cache 201, the process proceeds along the second path and the browser retrieves the content (step 260) from a source other than the rendered cache.

If the requested content is disposed in the rendered cache 201, but is determined to require updating (at step 230), then the process proceeds along the second path with the browser retrieving 260 the content from an updated source (e.g., the Internet 105). In some embodiments, where the updated source includes content formatted as MPEG, only the updated portion of the content is retrieved from the updated source.

The rendered content is then stored 320 in the rendered cache 201. In some embodiments, only the updated portion of the content is stored 320 in the rendered cache 201. Storing 320 and retrieving 260 only the updated portion of the content reduces the time and processor requirements for retrieving and storing content to update the rendered cache 201.

After the updated content has been stored 320 in the rendered cache 201 (as shown in Figure 3), the process continues along the first method path as long as the stored content does not become out of date. The first method path, as

shown in Figure 2, includes reading the semantic content and the rendered content 160, and restoring the semantic representation (step 240), to play 170 the rendered content for each request.

Figure 2A illustrates some embodiments of the methods for using multimedia data when the layout 140 process is performed by a server in communication with a client and is described in more detail below in the "Systems for Storing, Retrieving and Playing Multimedia Content" section.

Figure 3 illustrates a more detailed depiction of the layout 140 process shown in Figure 2 and also illustrates the storing 320 process using the rendered cache 201. After retrieving 260 the requested multimedia content with layout instructions and/or play instructions, the computer determines whether layout 140 is required for the multimedia content (step 310). The semantic representation 140B of the semantic features is generated during the layout 140 process, or generated from play instructions (step 315) when no layout is required.

After rendering 150, the rendered content 160 is stored 320 in the rendered cache 201. Similarly, after construction of the semantic representation 140B, the semantic representation is formatted as semantic content and also stored 320 in the rendered cache 201.

If the rendered cache 201 stored only the resulting rendered content 160, the description of the hyperlinks, display instructions and other semantic content would be lost. The semantic content can take the form of flat text files, XML or other structured files, or other proprietary formats. Some embodiments of the invention format the semantic content according to an XML language called VirtualModem™ Markup Language (VMML) to represent the semantic features of HTML pages and TVML presentations. The rendered content 160 and semantic content can be stored in a traditional cache, a database, a file system or other storage media. The underlying file system can be used to store the content in a directory and file hierarchy that represents the rendered cache 201.

The rendered content 160 stored 320 in the rendered cache 201 can include images, audio, text, full motion video, animations, etc. The content is stored in the rendered cache 201 regardless of its format [i.e. the rendered cache 201 can store binary large objects (blobs) or format-independent objects]. The format in which the semantic content is stored is independent of the rendered cache 201 mechanism. The content browsers and other client applications that access the rendered content 160 stored in the rendered cache 201 include processing resources adapted to recognize the format and interpret the semantic content appropriately.

According to some embodiments of the invention, content browsers and other client applications include processing resources to recognize and play 170 the rendered content 160 after the corresponding format-independent objects are retrieved 260 from the rendered cache 201. Some embodiments of the invention, including various VirtualModem™ presentation systems, can render HTML pages into a proprietary image format, called a fat macroblock (FMB), that is suitable for display on televisions. FMB's are described in greater detail by United States patent application serial number 09/287,235, entitled "System and Methods for Preparing Multimedia Data Using Digital Video Data Compression", filed April 6, 1999, having inventors Antoine Boucher, Paul E. McRae, and Tong Qiu, the entire contents of which are hereby incorporated herein by reference as if fully set forth herein.

In the case where the content is not missing but is outdated, the entire content can be retrieved 260, or just the outdated portions can be retrieved. By retrieving 260 only outdated portions some savings can be gained in the rendering 150 step by eliminating the need for a full rendering. For example, perhaps only an animated image on an HTML page has changed in the requested content. The rendering system can detect this situation and render 150 only the new animation rather than the entire page.

Once the needed portion of the request content has been retrieved 260, the content is rendered 150 before it is played 170. The retrieved content is handed to a rendering system that typically performs the following actions:

1. Laying out 140 of the content according to the appropriate rules (e.g., HTML rules).

2. Rendering 150 the content according to the rendering instructions 140A, thereby producing presentation data (e.g., for an MPEG image formatted as an FMB- or set of images for HTML frames) that represent the fully rendered representation of the content (e.g., the HTML page). The page may also have other graphical elements created for such things as animated GIFs.

3. Generating 315 a semantic representation 140B of the semantic features. Generally, the layout engine or the render engine creates the semantic representation 140B from the layout 140 or play instructions. For an HTML page, the semantic representation 140B can include the location, size, shape, and target of all HTML anchors (links to other HTML pages), the timing, location, and size of animated GIF frames, the size and relative location of HTML frames, information on HTML forms that can be accessed from the page, and HTML meta values.

4. Storing 320 the rendered content 160 [e.g., MPEG image(s)] in the rendered cache 201 using an appropriate index, e.g., a URI. The semantic content is also stored 320 in the rendered cache 201 using an appropriate index. In some embodiments, the semantic content can be stored 320 in an XML-based format so that it can be easily parsed and restored (e.g., in step 240) in the future. After the rendering system is finished, the rendered content 160 can be provided to the user by simply reading and restoring 240 and playing 170 the content.

The "format" of the semantic representation 140B is determined by the engine that generates the semantic representation (e.g. Netscape Communicator and Microsoft ® Internet Explorer use the DOM). This internal semantic representation 140B is then stored as a physical entity (semantic content) in the rendered cache 201. The format of semantic content is adapted for the browser engine that reads the semantic content for play 170. The format of the semantic content is sufficiently detailed for the browser engine to create its own semantic representation 140B. The semantic representation 140B in the browser engine

can be the same internal format that the layout/render engine uses or the semantic representation in browser can have a different format.

For some embodiments of the invention, the layout 140 process is performed by a server in communication with a client (e.g., set-top box) having rendering processing resources. For these embodiments a layout cache 318 can be coupled with the server to store rendering instructions and semantic content. These embodiments are described in more detail below in the "Systems for Storing, Retrieving and Playing Multimedia Content" section.

As shown in Figure 4, when a request is received for content already in the rendered cache 201 the rendering system process can be skipped entirely. The following is involved in playing 170 content already in the rendered cache 201.

1. Read the semantic content and the rendered content 160, and restore the semantic representation 140B from the semantic content stored in the rendered cache 201, e.g., the VMML description.

2. Play 170 the rendered content 160 on the user's screen according to this semantic representation 140B.

Some multimedia content, such as an HTML web page, does not fit entirely on a user's screen at once. For such partial page displays, the browser can use the semantic representation 140B to determine which portion of the page should be displayed, and for some embodiments which subset of the hypertext links are selectable on the page portion. An example of this scrolling is described below in the "HTML Page with Layout" example below.

Retrieving Content from the Rendered Cache

When a content browser, or other client application, requests a target index, such as a URI, the rendered cache 201 mechanism first looks in the rendered cache for a rendered representation of the content. The caching mechanism provides a means to search and retrieve this content based on the content's indexing indication. Examples of cache retrieval mechanisms include database queries, simple index files, file system directory structures, or traditional browser caches.

If the rendered content 160 can be found in the rendered cache 201, the content will be displayed very quickly. The semantic representation 140B of the rendered content 160 will be restored using the semantic content stored in the rendered cache 201 (i.e. the semantic features need not be computed again before the rendered content is played). For example, some embodiments restore the semantic representation 140B of a rendered HTML page by reading the VMML formatted semantic content.

If the content browser cannot locate a rendered representation of the multimedia content in the rendered cache 201, or the browser determines that the content is out of date, then the content can be retrieved 260 (either from a traditional cache 110, from the Internet 105, or from another content source) and rendered 150. The retrieval 260 and rendering 150 results in at least one new rendered cache 201 entry that can be used the next time the multimedia content is accessed.

A system that uses a rendered cache 201 will, after determining that no rendered representation is in the cache, perform the same steps as described above. That is, the HTML source will be read and the page laid out 140. The resulting rendering instructions 140A are followed but rather than displaying the page (or, alternatively, in addition to displaying) the rendering will be stored as a graphical image in the rendered cache 201. The semantic content (describing the location, size, and target URI of the single hyperlink on the image) is also stored in the rendered cache 201. The next time and every subsequent time the browser receives a request to view this URI, the browser simply reads the semantic content and the rendered content 160, restores the semantic representation 140B, and displays the rendered content. Thus, the use of the rendered cache 201 saves the cost of processing for layout 140, generation 315 of the semantic representation 140B, and rendering 150B. For more complicated HTML pages this savings can be substantial.

Examples

Specific embodiments of the invention are further described by the following, non-limiting examples which will serve to illustrate in some detail

various features of significance. The examples are intended merely to facilitate an understanding of ways in which the invention may be practiced and to further enable those of skill in the art to practice the invention. Accordingly, the examples should not be construed as limiting the scope of the invention.

HTML Page with Layout using a Distributed Server-Based Content System

Some embodiments of the invention provide storage 320, retrieval 260 and/or play 170 of HTML pages. One embodiment of the invention is represented by the Toronto Stock Exchange (TSE) HTML homepage illustrated by Figures 5A and 5B.

For this embodiment, the "content browser" can be broken up into a distributed server-based content preparation and viewing system. The viewing system can include a display device, e.g., a television, and a digital set-top box (such as a General Instruments DCT-2000).

For some HTML page embodiments, the set-top box has neither the processing nor the storage resources needed to render 150 or cache content. The set-top box typically does have the capability to decode and play MPEG images and Dolby AC-3 audio, and some limited graphics capabilities in order to do text and simple graphical overlays. For these embodiments, all access to rendering 150 processing resources and content stored in the rendered cache 201 is done at the server. These embodiments are described in greater detail in the "Systems for Storing, Retrieving and Playing Multimedia Content" section below.

In other embodiments, the set-top box, or other addressable processing equipment, can have processing resources and storage medium capable of rendering 150 and caching the content. In response to the server-based system receiving a request to view some content with the URI http://www.tse.com/ and determining that the content is either not in the rendered cache or is outdated, the server system browser requests retrieval of the TSE web page and any graphical elements the TSE web page references.

Once the web page and graphics have been retrieved 260 (either from a traditional cache 110 or from the Internet 105) the browser requests that the

content be laid out 140 and rendered 150. The rendering system creates an MPEG representation (in FMB format) of the rendered web page. Because MPEG is the only image format the GI DCT-2000 recognizes, we use MPEG in this example. The rendering system can also generate other FMB files representing animated GIF frames, if animated GIFs were referenced in the HTML page. The rendering system also creates a semantic representation 140B of the page including the location, shape, size, and target of all hyperlinks; location, size, and timing of animated GIF frames; HTML form information; and HTML meta information.

The FMB files are stored 320 in the rendered cache 201 using the URI of the HTML page ("www.tse.com") as an index. The semantic content is also stored 320 in the rendered cache 201 using the URI as an index. The semantic content is stored in an XML format called VMML. For distributed systems embodiments, e.g., the VirtualModem™ system, the internal semantic representation 140B for the layout/render engine is different than the semantic representation 140B for the browser engine (although these semantic representations 140B are conceptually equal). The stored semantic content (in the form of VMML for VirtualModemTM) is detailed enough to allow for "information transfer" so that two different semantic representations 140B can be used.

Once the rendered content 160 (FMBs) and semantic content (VMML) are stored 320 in the rendered cache 201, the browser can then read and restore the semantic representation 140B based on the VMML file. Using this semantic content the web page can be displayed.

The first screen capture (Fig. 5A) of the TSE homepage shows the top portion of the page. The rectangular highlight box 510 in the top left corner indicates that the user can select the first hyperlink for viewing. Users can press arrow keys on their remote control to move from one link to another link on the page. The browser provides enough information for the set-top box to draw the highlight box 510 and to navigate the page from link to link using the arrow keys.

Eventually the user may scroll past the bottom of the screen. The set-top will then inform the server-based browser that a scroll is required and the browser will then determine from the semantic content which new portion of the rendered MPEG should be visible and which new subset of the hyperlinks is now selectable.

The second screen capture (Fig. 5B) illustrates the TSE homepage after a scroll down. The user can continue to view the same page scrolling around and viewing the content in the fashion described above. However, once a link is selected, the browser is informed of the corresponding new URI request and the retrieval 260 (or read and restore 240) process are initiated again after the browser receives a play request 120.

HTML Page with Layout using a Self-Contained Content System

Some embodiments do not use the distributed server-based content preparation and viewing system described in the above example (HTML Page with Layout). Instead, these embodiments are self-contained content systems with layout 140, rendering 150, and play 170 processes all combined in a single computer program. Some of these embodiments use a Netscape Communicator browser, a Microsoft ® Internet Explorer browser, or a Spyglass HTML browser residing in a Scientific Atlanta Explorer™ Model 2000 home communications terminal (i.e., set-top box).

Such desktop and set-top browsers can also use the invention to reduce retrieval 260 and playing 170 time, and to decrease processor usage. Traditional web browsers have long used caching technologies to minimize the need to use slower content retrieval 260 methods such as network access. These browsers store the original retrieved content in a cache database. When a request to view content is received, the browser searches the traditional cache 110. If the content is not in the cache, then the browser retrieves the content from an alternate source (such as the Internet 105). Visiting web sites that reside in a rendered cache 201 results in almost instantaneous display of the web site content rather than the usual delay (due to the cost of layout, rendering and creation of semantic context) that is normally seen.

Whether or not the content was found in the traditional cache 110, the content is then read and laid out 140 according to the rules of HTML. Laying out 140 produces rendering instructions 140A and a semantic representation 140B of the content. The page is then rendered 150 to a graphical format (typically a bitmap) and played 170 according to the semantic representation 140B. These steps are performed each and every time the content is requested.

For the self-contained embodiments of the invention, when a request for content is received the browser will search in the rendered cache 201 to determine whether a rendered representation of the content is available. If the content is not in the rendered cache 201, or if the rendered content is found to be outdated, then the content must be requested from an alternate source (such as the Internet 105, or a traditional cache 110). Once the content is received, it will go through the same layout 140, rendering 150, and generation 315 of the semantic representation 140B steps as these browsers do now.

The difference is that once the rendering 150 and generation 315 of the semantic representation 140B is complete the rendered content 160 and the semantic representation 140B are stored in the rendered cache 201.

Once the content is stored in the rendered cache 201, then each time the browser receives a request for this content, the browser simply reads and restores 240 the semantic representation 140B and plays 170 the rendered content 160 according to this semantic representation. The format of the rendered content and semantic representation are entirely up to the browser. It is recommended that the rendered content be stored in a "native format." That is, a format that the browser can immediately recognize and does not have to convert to a recognized format. It is also recommended that the format for the semantic representation 140B be rich enough to cover all the various semantic elements that HTML can describe. VMML is a good example of such a format. For self-contained systems the format of the internal semantic representation 140B is likely to be the same for both the layout 140/render 150 and browser portions of the program.

Another related embodiment that could benefit from an embodiment of the invention is what is commonly referred to as a "web proxy." A web proxy

is a computer program that retrieves content on behalf of content browsers. Various embodiments of the invention enable the web proxy to only retrieve 260 content from the Internet 105 for the first request, while all future requests for the content from browsers using the proxy use the locally cached version.

Note that in either the distributed or self-contained scenarios, the task of converting to and from the stored semantic content format is up to the relevant engines (layout 140/render 150 engine for storing and browser engine for retrieval 260).

An alternate scenario could involve the layout/render engine transferring the semantic representation (through some communications medium) to a "rendered cache server" that converts the representation into semantic content. This server would also receive request to retrieve content from the cache and would read the semantic content, convert it to an appropriate internal representation and then transfer this representation. In this case the task of converting to and from semantic content is entirely up to the "rendered cache server". In practice, this approach is less flexible than alternative approaches.

In the case where the web proxy and the content browsers all have access to the same storage or have access to a fast internal communications network, the web proxy could perform the layout 140, rendering 150, and generation 315 of semantic representation 140B steps on behalf of the content browsers. In such a scenario, when a content browser receives a request for content, the content browser can either look directly in the rendered cache 201 or query the web proxy for the rendered content 160. The browser can then simply read and restore 240 the semantic content and display the rendered content 160 accordingly. This use of the web proxy allow for the use of very small and efficient web browser implementations since all the resources for layout 140, rendering 150, and generation 315 of the semantic representation 140B are external of the browser.

An intelligent web proxy can pre-render the content that it downloads in order to offset the rendering cost in browsers. This approach is especially beneficial in situations where client computing resources are limited. A key application of this approach is in the emerging market of set-top devices and

other network computers. These devices typically have tightly constrained resources and do not presently provide true web browsing. The use of a rendered cache proxy would offload the process of layout 140, rendering 150, and generation of the semantic representation 140B.

In some embodiments, word processing programs can store 320 rendered documents in a rendered cache 201 for faster loading and previewing. Using the rendered cache 201 for storing 320 word processing documents also enables programs other than the word processor to preview the content without using proprietary plug-ins or libraries.

HTML Page with Layout using a More Capable Settop Client

For some embodiments of the invention, the set-top box has the processing resources to render HTML content and the storage resources to store rendered content. Once a web page and its graphics have been retrieved (either from a traditional cache or from the internet) the layout engine will lay out the content creating a paint stream. The paint stream describes how to render the page and where the interactive elements are (e.g. hyperlinks, form elements, animated gifs, etc.).

For example, for http://www.tse.com/ (introduced in an earlier example), the server, after retrieving the HTML and images, lays out the page and transmits the entire paint stream to the settop box. The settop box then renders the page according to the render instructions and then stores the rendered image and the semantic content on a local storage device (either disk or memory). It then displays the top portion of the page (e.g., Fig. 5A). The semantic information in the paint stream allows the settop to highlight the hyperlinks.

Eventually the user may scroll past the bottom of the screen and the settop will display a new portion of the rendered image (e.g., Fig. 5B) allowing the user to navigate a different portion of the HTML page. If the user returns to the http://www.tse.com/ web page, the settop can then simply display the version in its local cache and restore the semantic information from the semantic content stored in the local cache.

Often the rendered representation of a web page may take more storage capacity than the paint stream. According to one embodiment of the invention, as an alternative to storing the rendered image in the local rendered cache, the settop could store the paint stream itself. The settop has the option of rendering only the portion of the web page that is currently visible, or it may render the entire page. Once the user returns to the web page, the paint stream is restored from the cache and re-rendered. This helps to obviate the need for the layout step.

According to another embodiment of the invention, in the case where the settop has the processing resources to quickly render web pages but lacks the storage capacity to contain a local cache, the server can contain a cache of paint streams. In this scenario, after the layout engine is finished laying out http://www.tse.com/, the paint stream is transmitted to the settop and is also stored in a server cache. The next time the client (or any other client in communication with the server) requests http://www.tse.com/, the layout step can be skipped and the cached paint stream can be transmitted to the client. This helps to allow the layout step to be skipped.

Multimedia Content with Play Instructions

Some embodiments of the invention provide storage 320, retrieval 260 and/or play 170 of multimedia content. The multimedia content can include images, audio, text, graphics, and full motion video, all of which can be timed to play at different moments. This multimedia content can have a means of referencing other multimedia content in a manner similar to HTML hyperlinks. Some embodiments of the invention, including the VirtualModemTM system from Interactive Channel, use an XML language called TVML to represent the play instructions of a multimedia presentation. TVML can include markup to represent the following play instructions of the multimedia content:

1. Timing of multimedia content playing;
2. Order of multimedia content playing;
3. Size and location of multimedia content; and

4. Location, size, shape, and target URI (or other index) of hyperlinks.

One embodiment of the invention is represented by the News Menu TVML presentation illustrated by Figs. 6A through 6D. Fig. 6A illustrates a timeline representing how the News Menu TVML presentation should be played. Figs. 6B through 6D show the images that make up the News Menu TVML presentation. As in the previously described embodiment (HTML with layout) the "content browser" can be broken up into a distributed server-based content preparation and viewing system.

The server-based system can receive a request to view some content with the URI http://www.virtualmodem.com/news.tvml and then determined that the content is either not in the rendered cache 201 or is outdated. The browser can respond to this circumstance by submitting a request to retrieve the TVML presentation and any multimedia elements referenced by the presentation. Once the presentation and its multimedia elements have been retrieved 260 (either from a traditional cache 110 or from the Internet 105), the browser requests that the content be rendered 150. In this case, layout 140 is unnecessary and the rendering 150 can be limited to converting the multimedia content into a format that the set-top recognizes. In the case of the GI DCT-2000 images and full motion video are converted to MPEG formatted data and audio is converted to Dolby AC-3 formatted data.

The rendering system can also generate 315 a semantic representation 140B of the page from the TVML play instructions. The semantic representation 140B can include context such as the relative play times and order of the multimedia content; the location, shape, size, and target of all hyperlinks; and TVML meta information. The rendered content 160 can be stored 320 in the rendered cache 201 using the URI of the presentation ("www.virtualmodem.com/news.tvml") as an index. The appropriately formatted semantic content based on the semantic representation 140B is also stored in the rendered cache 201 using the URI as an index. For some embodiments of the invention, the semantic content is stored in a VMML format.

Once the rendered content 160 and semantic content (VMML) are stored in the rendered cache 201, the browser can read and restore 240 the semantic representation 140B from the VMML file in which the semantic content is disposed. Using this semantic representation 140B the presentation can be displayed.

Figure 6A shows the start time and duration that each image of the presentation should be played, and illustrates the start time and duration of the accompanying audio. The presentation plays 170 from t_0 to t_3.

Figures 6B through 6D show each of the images used in the for the News Menu TVML presentation. The first image 650 of the presentation, shown in Fig. 6B, includes a single circle with a small diameter around the top of the transmitter to indicate that a signal is being sent from a transmitter. As shown by the first time line 610, the first image 650 is shown from t_0 to t_1.

The second image 660 of the presentation, shown in FIG. 6C, includes three circles around the top of the transmitter to indicate that the signal will be received by the user sooner than when the first image 650 was displayed. As shown by the second time line 620, the second image 660 is shown from t_1 to t_2.

The third image 670 of the presentation, shown in FIG. 6D, includes a first hyperlink that is enclosed by a rectangular highlight box 510 to indicate that the first hyperlink "World News Update" is presently available for selection. As shown by the second time line 630, the third image 660 is shown from t_2 to t_3. A user can press arrow keys disposed on the user's remote control devices to move from link to link in the third image 670. The browser provides enough information for the set-top box to draw this rectangle and to navigate using the arrow keys from link to link. If the multimedia content is larger than the physical screen then it becomes possible to scroll in the same manner as described in the "HTML with layout" example. As shown by the fourth time 640, the accompanying audio plays 170 for the entire duration of the News Menu TVML presentation.

In the News Menu TVML presentation example only the third image 670 of the presentation contains hyperlinks. However, in other embodiments of the invention, any of the earlier images can also contain hyperlinks. The

browser can update the client (set-top box or other addressable processing equipment) whenever the semantic representation 140B (e.g. hyperlink information or image display duration) changes.

The presentation can play 170 until all multimedia objects have been played. The user can continue to view the last image of the presentation in the same manner as for HTML pages. The user can also manipulate the remote control VCR functions to rewind, fast-forward, or pause the presentation. However, once a hyperlink is selected, the browser will be informed of the new URI request and the content retrieval process will start again with a request for content.

Multimedia Content with Play Instructions using a More Capable Settop Client

Some embodiments of the invention provide storage, retrieval, and play of multimedia content. In the case where the settop is more capable, the server-based layout engine can transmit a paint stream consisting of the play instructions and multimedia content to the settop. Upon receiving this paint stream, the settop can then play the multimedia presentation according to the play instructions in the paint stream. This kind of paint stream may not require any rendering at all. The paint stream can be cached locally in the settop or on the server.

It is sometimes not possible to send all the multimedia content for such a presentation to the settop at once. According to one embodiment of the invention, in such a senario the paint stream can consist only of the play instructions. Upon receiving the play instructions, the settop can then make requests to the server to transmit the appropriate multimedia content for the portion of the presentation currently being played.

Systems for Storing, Retrieving and Playing Multimedia Content

Some embodiments of the invention include systems for storing 320, retrieving 260 and playing 170 multimedia content using a rendered cache 201. Listed below are the key elements of a system that can implement various embodiments of the invention. Previous descriptions and examples, mentioned

in the "HTML Page with Layout using a Distributed Server-Based Content System" section, have illustrated the use of the invention in a distributed server-based system. In such a system the various complimentary components, such as those listed below, are typically found in separately running processors that can reside in a single computer or in multiple connected computers. Some embodiments, such as the VirtualModem™ system can include the following components:

Web crawler processing resources adapted to access multimedia content from source data storage. The multimedia data can include HTML and TVML content. The source data storage can include at least one of the Internet 105 and a web proxy cache.

Rendering processing resources adapted to generate semantic representation 140B of, and render 150 multimedia data, and can format the semantic representation as semantic content. In some embodiments, a rendering program can also be adapted to layout 140 the multimedia data.

Multimedia playing processing resources, such as an audio/video terminal server (AVTS), adapted to play multimedia content. Such play can include displaying images and playing audio and full motion video. Some embodiments of an AVTS are described in greater detail in United States patent application serial number 09/255,052, entitled "System and Method for Interactive Distribution of Selectable Presentations," filed February 22, 1999, and having inventors: Antoine Boucher, James Lee Fischer, and Allan E. Lodberg, the entire contents of which are hereby incorporated herein by reference as if fully set forth herein.

Browser processing resources adapted to interpret the semantic content and control when and how the multimedia content should be played. The browser processing resources can act as the "control center" for the entire process. The browser processing resources can communicate with the web crawler, rendering, and the multimedia playing processing resources and coordinate the interactions of each of these.

For some embodiments of the invention, a server-based system can be used to perform the layout 140 step only. The render 150, play 170, and store

320 steps can be performed by client addressable processing equipment (e.g., a set-top box) in communication with the server based system.

Methods for using multimedia data according to these embodiments are represented by Figure 2A. After the server retrieves 260 the requested content, the server lays out 140 the content and thereby generates rendering instructions 140A and a semantic representation 140B of the multimedia content. The combination of the rendering instructions 140A and the semantic representation 140B can be referred to as a paint stream 145.

The paint stream 145 is then transmitted to at least one client in communication with the server. The client can be a set-top box, or other addressable processing equipment (APE). Upon receipt of the paint stream 145, the client processing resources can render 150 the multimedia content. Embodiments of the invention for which the rendering processing resources and a client rendered cache 201 are disposed at the client can provide more rapid play 170 of the multimedia content stored 320 in the client rendered cache. This rapid play 170 is provided by avoiding the time required to transmit the request for content to the server, and the time to transmit the data corresponding to the requested multimedia content from the server to the client.

The layout 140 step generates a semantic representation 140B and a set of rendering instructions 140A for the multimedia content. The semantic representation 140B and the rendering instructions 140A can be transmitted (via a network) to the client set-top box in a paint stream 145. The rendering instructions 140A and semantic representation 140B can be sent separately or can be bound together. For some of these embodiments, the rendering instructions 140A can include the multimedia elements, e.g., bitmaps, audio, and graphics. In other embodiments, the rendering instructions do not include the multimedia elements, and the multimedia elements can be requested by the client set-top box separately from the request for the paint stream data.

In some embodiments where no layout 140 is required, the server-based system can generate 315 the semantic representation 140B from the play instructions. For these embodiments, the paint stream 145 can include only the semantic representation 140B derived from the play instructions.

After receiving the paint stream 145, the client set-top box can then render 150 the multimedia content and play 170 the multimedia content according to the semantic representation 140B.

In some embodiments, the set-top box can include processing resources to store 320 the paint stream 145 data on a local storage device adapted to store such data for rapid reading, rendering 150 and playing 170. The cache adapted to store 320 the paint stream 145 data is also referred to herein as a "paint stream cache" or as a layout cache 318 as shown in Figure 3. The set-top box can also include processing resources to render 150 the paint stream 145 data and play 170 the multimedia content.

Alternatively, the set-top box can include processing resources to render 150 the paint stream 145 data, and then store 320 data corresponding to the rendered representation along with the semantic representation 140B portion of the paint stream in a rendered cache 201 disposed at the set-top box. The rendered representation stored in the set-top box rendered cache 201 is generated from the paint stream 145 by the rendering 150 process performed by the set-top box. The set-top box rendered cache 201, and/or layout cache 318 can be a hard disk, another re-writeable storage medium, or a computer memory.

In some embodiments, the server-based system can store 320 data corresponding to the paint stream 145 in a layout cache 318. The server-based system can read the data stored in the layout cache 318 corresponding to a request from the client, and then transmit data corresponding to the paint stream 145 to the client. The client can then forward the data corresponding to the paint stream 145 to the rendering processing resources which render 150 the content. The client can then play 170 the content.

For another set of alternative embodiments, the set-top box can receive the paint stream 145 data, render 150 the paint stream 145 data, store the rendered content 160 and the semantic content, read the rendered content and the semantic content and restore the semantic representation (step 240), and play 170 the content. The next time the user requests the content, the client set-

top box can play 170 the content without passing on the request to the server based system.

FIG. 7 illustrates the components and features configured in a system for accessing multimedia content 700 using a rendered cache representing one embodiment of the invention. The system for accessing multimedia content 700 using a rendered cache includes the components and features described below, including: access to source content 710, at least one layout engine 720, at least one render engine 730, a rendered cache 201, at least one multimedia play engine 750, at least one browser engine 760, and a display 770. These components can be combined together to form one or more computer programs that implement the storing 320, retrieving 260 and playing 170 methods described above.

Source content is content that is not yet rendered. The source content can include HTML, XML, images, audio, text, and full motion video. Access to source content 710 can be through an Intranet, the Internet 105, a web proxy, or on local storage. Connections adapted to provide such access can be through any carrier capable of providing sufficient bandwidth for practical retrieval 260 the content, such as: digital subscriber line (DSL), cable modem, T-1, T-2, T-3, OC-1 through OC-256, fiber distributed data interface (FDDI), E1 through E5, Ethernet, fast Ethernet, and Gigabit Ethernet. Access to source content 710 can also include processing resources adapted to use standard Internet protocols such as TCP/IP and HTTP, and to read files from a file system. The component providing access to source content 710 includes processing resources for retrieving the source content, such as the content fetch 715 resources shown in FIG. 7.

The system for accessing multimedia content 700 using a rendered cache can include layout processing resources, such as a layout engine 720, adapted to derive rendering instructions 140A from a content definition (e.g., HTML). The layout engine 720 can also derive a semantic representation 140B of the features of the content from the layout 140, or from the play 170 instructions. Netscape ® Communicator and Microsoft ® Internet Explorer both contain processing resources to perform HTML layout 140 as part of their overall functionality.

Stand-alone layout engines 720 include Spyglass Device Mosaic, NGLayout from Mozilla, and Chimera. In some embodiments, processing resources other than the layout processing resources can be adapted to generate 315 the semantic representation 140B from play 170 instructions.

The system for accessing multimedia content 700 using a rendered cache can include rendering processing resources, such as a render engine 730, adapted to create a graphical representation of content that has been laid out 140 by the layout engine 720. The rendering engine 730 can also have the capability of converting content that does not require layout 140 into a form that is ready for rapid play 170.

Many layout engines 720 also include a render engine 730. Systems that have layout engines 720 that do not include a rendering 150 capability have a separate render engine 730 and typically specify the interface that a rendering engine must have. Both Netscape Communicator and Microsoft® Internet Explorer include rendering engines as part of their overall functionality. Both of these browsers render 150 the content into a bitmap that can be displayed on a computer monitor display 770. Some embodiments, such as the VirtualModem™ system, use their own custom render engine 730 that renders the content to MPEG files stored in FMB format.

The rendered cache 201 provides access to an indexed storage mechanism. The rendered cache 201 stores both the rendered content 160 and the semantic content so that these data objects can be easily retrieved 260 at a later time. The rendered cache 201 includes as indexing mechanism that can take a variety of forms including database queries, index files, file system directories.

The format of the rendered content 160 is independent of the storage mechanism. A format that requires little or no conversion at play 170 time (i.e. a "native format") provides greater time and processing savings using the rendered cache 201.

The format of the semantic content is also independent of the storage mechanism. The semantic content format used in the system for accessing multimedia content 700 using a rendered cache that fully captures all the

semantic features of the rendered content 160 provides enhanced play results. The semantic content format can avoid unneeded complexity to ensure that the processing and time required to restore the semantic representation 140B are less than that required to layout 140 and re-render the content.

The system for accessing multimedia content 700 using a rendered cache includes multimedia play processing resources, such as a multimedia play engine 750, adapted to play the rendered content 160 on a display 770 device. The multimedia play engine 750 can read the rendered content 160 directly from the rendered cache 201 indexed storage mechanism, read the rendered content from memory, or otherwise receive the rendered content from an external source. Netscape Communicator and Microsoft® Internet Explorer both contain, as part of their overall functionality, processing resources to display multimedia content to a computer display 770. Some embodiments, including the VirtualModem™ system, include a separate program that is part of the overall distributed system, called the AVTS, that is adapted to play multimedia content to set-top boxes or other addressable processing equipment (APE). In some embodiments of the invention, the set-top box (or other APE) includes a processing unit capable of performing computational tasks similar to those performed by a desktop computer. The set-top box (or other APE) can also include computer memory for storage of computer programs and data. References to "computer" in this document, can therefore be applied to the set-top boxes and APE of these embodiments.

The system for accessing multimedia content 700 using a rendered cache also includes a browser engine 760 adapted to interpret the semantic representation 140B of the rendered content 160 being played 170. The browser engine 760 can read the semantic content directly from the rendered cache 201 indexed storage mechanism, or interpret the rendered content 160 from memory, or otherwise receive the semantic content from an external source.

The browser engine 760 can be adapted to interpret the semantic features from the semantic content. In some embodiments, the browser engine 760 is adapted to control navigation of hyperlinks (i.e. determining from user

input which content should be displayed next). The browser engine 760 also can determine which portions of the rendered content 160 should be played 170, and which corresponding portions of the semantic representation 140B are active (e.g., when scrolling an image).

The browser engine 760 can be included in commercially available software such as Netscape Communicator, Microsoft ® Internet Explorer, or any other browser engine that is adapted to perform the functions described above. Netscape Communicator and Microsoft® Internet Explorer both contain, as a part of their functionality, processing resources adapted to interpret a semantic representation 140B [or Document Object Model (DOM) as both call it]. Both of these browsers use the DOM to determine which links are currently visible (and which others are scrolled off the screen), animated GIF timing and location, information about HTML forms and other HTML features. Some embodiments, such as the VirtualModem™ system, include a browser program that coordinates the retrieving 260 of content, layout 140 and rendering 150 of content, and playing 170 of rendered content. These browser embodiments can also contain processing resources for reading semantic content from the rendered cache 201 and restoring the semantic representation 140B.

The above engines (layout, render, play, and browser) are all at least loosely coupled. That is, they need not be part of the same program but there needs to be some form of communication between them all. This communication can take a variety of forms including inter-process communication (such as shared memory, pipes, or messaging protocols), or shared files. Some embodiments, such as the VirtualModem™ system, use a communications protocol built on a user data protocol (UDP) to communicate between the various engines. Netscape Communicator and Microsoft® Internet Explorer include all the engine components in the same program.

There is no requirement that any of the above system components be directly tied together (i.e. included in the same program). However, there are advantages to tightly coupling certain components. For example, it is more efficient to couple the layout engine 720 and the render engine 730 in the same

program. In such a scenario the rendering instructions 140A resulting from layout 140 process can be used directly by the rendering engine 730 component. If the layout engine 720 and the render engine 730 are separate programs, then some intermediate form of rendering instructions (e.g. either a file or data passed over a network) can be used.

The component responsible for accessing source content 710 includes processing resources to access the communications carrier and the underlying communications protocol. It is not required that the other engine components have these processing resources.

The layout engine 720 and the render engine 730 have access to the rendered cache 201 storage mechanism since they read the rendered content 160 and the semantic content.

The multimedia play engine 750 has access to at least the rendered content 160 portion of the rendered cache 201 storage mechanism. The browser engine 760 has access to at least the semantic content portion of the rendered cache 201 storage mechanism. Both the multimedia play engine 750 and the browser engine 760 can have full access to the entire rendered cache 201 storage mechanism but at minimum they have access to their respective content.

Splitting access to the rendered content 160 and the semantic content enables efficient distribution of the multimedia play engine 750 and browser engine 760.

Figure 7A illustrates a paint stream system 701 for accessing multimedia data, representing one embodiment of the invention. In the paint stream system 701, the render engine 730 is disposed at the client 725. The server lays out 140 the content and then transmits the paint stream 145 data to the client 725 for rendering 150 and playing 170.

Figure 7B illustrates a self-contained system 702 for accessing multimedia data, representing one embodiment of the invention. For this embodiment, all system components reside at a single location such as the client 725.

Cable System

Fig. 8A illustrates the location and connection of various components of a rendered cache for embodiments with more capable settops. In this drawing server 812 is connected to Internet 810 and performs the content fetch and layout steps. The server resides at cable system headend 800. The result of the layout step is a paint stream and semantic content. The paint stream and semantic content are transmitted via cable connection 818 to settop 830. Settop 830 is coupled to display 842, which may be standard television analog, digital, HDTV, LCD, computer monitor, or other display or monitor. Although not shown in this figure, the headend server could contain a server-side rendered cache that stores the paint stream and semantic content. This could help reduce latencies associated with fetching and laying out the content.

Settop 830, which resides at consumer's premises 840 (or other user location), has the processing resources to efficiently render (render engine 824) the content based on the paint stream instructions sent from server 812 residing at cable company headend 800. The settop also has storage capacity to store paint stream 820 and the semantic content 822 in a local rendered cache 844 in order to reduce latencies associated with communication to server 812. Settop 830 also has the processing and graphical resources to play (play engine 826) the rendered content and to perform browser functionality (browser 826).

Figure 8B illustrates the location and connection of various components of a rendered cache for embodiments, for example, with less capable settops. In this drawing server 812, which resides at cable system headend 800, is connected to Internet 812 and performs content preparation and browsing steps. Rendered cache 844 also resides on server 812. Settop 830, which resides at consumer's premises 840 or other user location, is used merely as a display device for the rendered content. Functions of content fetch 814, layout engine 816, render engine 824, play engine 826, and browser engine 828 are performed by server 812.

Fig. 8C illustrates the location and connection of various components involved in the rendering process, including a partial render engine, according to an embodiment of the invention. Although Fig. 8C shows an embodiment of

the invention in the context of a cable system, the principles described may apply to other types of communication systems. For some embodiments of the invention, the layout process and some of the rendering process can be performed by a server 812 in communication with a client (e.g. set-top box 830) having at least some rendering processing resources, e.g. partial render engine 850.

In an embodiment such as shown in Fig. 8C, server 812 first lays out newly retrieved multimedia content (either from a traditional cache or from Internet 810) to form rendering instructions and a semantic representation of the multimedia content. Partial render engine 852 in server 812 renders at least some of the multimedia content according to at least some of the rendering instructions. The rendered content, the remaining rendering instructions, and the semantic representation are then transmitted to client (e.g., set top box 830). After detecting the rendered content, rendering instructions, and semantic representation, the client can then complete the rendering using local processing resources (e.g., partial render 850) resulting in fully rendered multimedia content. The fully rendered multimedia content is then stored in a local rendered cache 844 and played according to the semantic representation. In subsequent access to this content, the semantic representation is read and the rendered content is played according to the semantic representation

In the system according to the above text and Fig. 8C, a number of alternative scenarios are possible:

1. The client stores the partially rendered image, remaining rendering instructions, and the semantic content in the local rendered cache. In subsequent access to this content, the remaining rendering processing is performed before playing the multimedia content according the semantic representation. This scenario can be used when the fully rendered image is too large to store in the client rendered cache.

2. The server stores the partially rendered image and transmits only the remaining rendering instructions and semantic content to the client. When the client browser needs to display the multimedia content, it requests the partially rendered portion of the multimedia content be transmitted and then the

client performs the remaining rendering steps along with playing the transmitted rendered content according to the semantic representation. This scenario could be used when the client does not have enough storage resources to store the partially rendered content.

3. The server stores the partially rendered content, the remaining render instructions, and the semantic content in a server-based rendered cache. When the client makes a request for the content, the server can transmit the partially rendered content, remaining rendering instructions and the semantic content. This avoids the layout process and part of the rendering process. This scenario could be used when the client does not have enough storage resources to store any of the partially rendered content, remaining rendering instructions, and the semantic content.

In a system such as that illustrated in Fig. 8C, the determination of how much of the rendering process is performed by the server and how much is performed by the client may depend largely on the rendering processing resources of the client.

The following is an example of accessing an HTML page with a system having a partial render engine such as that shown in Fig. 8C and described above. In response to the browser process requesting to view some content with the URI http://www.tse.com/ and determining that the content is either not in the rendered cache or is outdated, the server based portion of the system requests retrieval of the web page and graphical elements the TSE web page references. Once the web page and graphics have been retrieved (either from a traditional cache or from the Internet), the content is laid out according to the rules of HTML.

Continuing with the example, the server-based portion of the render process then renders all elements of the page with the exception of the text. That is, all graphics and other non-text elements are rendered in the locations that the layout process previous determined. The rendered image, the remaining rendering instructions (providing the text, locations, colors, and font sizes), and the semantic representation are all transmitted to the client. The client has processing resources to detect this content and display the rendered image and

render the remaining text on top of the image. Scrolling can be accomplished by displaying the next portion of the image and rendering the next portion of the text. The client stores the fully rendered image and semantic representation in a local rendered cache or it stores the partially rendered image along with the remaining rendering instructions and the semantic representation in a local rendered cache. If the client does not have the resources to store all these content elements, it can either store a portion (e.g. only the rendering instructions and the semantic representation) or none of the elements. In either case the server stores the content elements that the settop cannot in a server-based rendered cache.

Formatting the Semantic Content

Some embodiments of the invention use an eXtensible markup language (XML) language to format and store 320 semantic content in the rendered cache 201. Embodiments including the VirtualModem™ system use a markup language called VMML to format and store 320 semantic content in the rendered cache 201.

VMML contains elements to describe the semantic features of both HTML and TVML. TVML is another XML language originally based on synchronized multimedia integration language (SMIL) from the World Wide Web Consortium at http://www.w3.org/. The descriptive elements include:

1. Multimedia elements - The <img>, <audio>, <video>, and <text> elements are used to describe fully rendered multimedia objects. The <screen> element is used to describe fully rendered HTML. Each of these elements can include an optional start time using the "begin" attribute.

2. Aggregation elements - The <par> and <seq> elements are used to describe how the multimedia elements are played. Elements inside a <par> are played in parallel. The start times of multimedia elements in a <par> are relative to the beginning of the <par>. Elements inside a <seq> are played sequentially. The start times of multimedia elements in a <seq> are relative to the end of the previous element. Both the <par> and <seq> elements can define optional start times using the "begin" attribute.

For example, the following <par> element contains an <audio> and two <img> elements which are played in parallel (i.e. at the same time). The display of the second image is delayed by 5 seconds.

```
<par>
    <audio src="voice-over.ac3"/>
    <img  src="first-screen.fmb"/>
    <img  src="second-screen.fmb" begin="5.0s"/>
</par>
```

3. HTML elements - The <screen> element is used as a container for all the semantic information concerning a rendered HTML page. Elements allowed in a <screen> element include:

<frame> - contains attributes for defining the FMB (the rendered frame), size, and location relative to other frames of the HTML page;

<anchor> - each <frame> element can contain a list of <anchor> elements which describe the location, size, shape, and target of HTML hyperlinks;

<form> - each frame can contain form elements which fully describe HTML forms;

<animation> - <frame> elements can contain animation elements that describe the timing, size and location of animated GIFs.

4. Non-display elements - The <title> and <meta> elements describe non-audiovisual features of the content. Examples of <meta> information include HTML refreshes, and expire metas.

5. Anchors - Information about non-HTML hyperlinks is also described in VMML <anchor> elements.

6. Applets - The <applet> element instructs the browser to run other applications.

VMML contains other minor elements and a wide variety of attributes but the above list describes the major features. VMML is capable of describing all the various features of TVML and HTML in sufficient detail that the semantic representation 140B can be reconstructed after reading the semantic content from the rendered cache 201. The reconstruction of the semantic representation 140B includes simple tokenization (i.e. text parsing) using freely

available tools such as sgml-lex (available from http://www.w3.org/). The parsing process is much faster and uses far fewer processor resources than the processes of layout 140 and rendering 150.

The term coupled, as used herein, is defined as connected, although not necessarily directly, and not necessarily mechanically. The term substantially, as used herein, is defined as approximately (e.g., preferably within 10% of, more preferably within 1% of, most preferably within 0.1% of).

Advantages of the Invention

A rendered cache 201 coupled with multimedia content render, play, and browser processing resources, representing an embodiment of the invention, can be cost effective and advantageous for at least the following reasons. The rendered cache 201 enables the play 170 of multimedia content in less time and using less data processing because the steps of layout and rendering are eliminated.

All the disclosed embodiments of the invention described herein can be realized and practiced without undue experimentation. Although the best mode of carrying out the invention contemplated by the inventors is disclosed above, practice of the invention is not limited thereto. Accordingly, it will be appreciated by those skilled in the art that the invention may be practiced otherwise than as specifically described herein.

For example, although the rendered cache 201 described herein can be a physically separate module, it will be manifest that the rendered cache 201 can be integrated into the apparatus with which it is associated. Furthermore, all the disclosed elements and features of each disclosed embodiment can be combined with, or substituted for, the disclosed elements and features of every other disclosed embodiment except where such elements or features are mutually exclusive.

It will be manifest that various additions, modifications and rearrangements of the features of the invention may be made without deviating from the spirit and scope of the underlying inventive concept. It is intended that the scope of the invention as defined by the appended claims and their

equivalents cover all such additions, modifications, and rearrangements. The appended claims are not to be interpreted as including means-plus-function limitations, unless such a limitation is explicitly recited in a given claim using the phrase "means for."

CLAIMS

What is claimed is:

1. A method, implemented in at least one computer, for storing multimedia data, comprising:

detecting multimedia content, the multimedia content including play instructions and at least one multimedia element, the at least one multimedia element including at least one of graphical images, audio, text, and full motion video;

generating a semantic representation and rendering instructions for the multimedia content from the play instructions, the play instructions including at least one of timing of the multimedia content and ordering of the multimedia content, the semantic representation describing at least one of: characteristics of a rendered representation of the multimedia content, and relationships between different multimedia elements disposed in the rendered representation;

storing data corresponding to the rendering instructions in a paint stream cache; and

storing data corresponding to the semantic representation in the paint stream cache.

2. The method for storing multimedia data of claim 1, wherein:

the paint stream cache is disposed at a client, the client is adapted to communicate with a server;

the server performs the detecting and generating steps; and

the method includes prior to storing the data corresponding to the rendering instructions:

the server transmitting data corresponding to the rendering instructions to the client; and

the server transmitting data corresponding to the semantic representation to the client.

3. The method for storing multimedia data of claim 1, wherein the paint stream cache is disposed at a server, and the server is adapted to communicate with at least one client.

4. The method for storing multimedia data of claim 1 including prior to storing the data corresponding to the rendering instructions:

formatting the rendering instructions into data formatted for rapid reading from the paint stream cache; and

formatting the semantic representation to form semantic content, the semantic content formatted for rapid reading from the paint stream cache, and rapid restoration into the semantic representation.

5. A method, implemented in at least one computer, for storing multimedia data, comprising:

a server detecting multimedia content, the multimedia content including play instructions and at least one multimedia element, the at least one multimedia element including at least one of graphical images, audio, text, and full motion video;

the server generating a semantic representation and rendering instructions for the multimedia content from the play instructions, the play instructions include at least one of timing of the multimedia content and ordering of the multimedia content, the semantic representation describes at least one of: characteristics of a rendered representation of the multimedia content, and relationships between different multimedia elements disposed in the rendered representation; and

the server transmitting data corresponding to the semantic representation and the rendering instructions to a client.

6. A method, implemented in a computer, for storing multimedia data, comprising:

detecting paint stream data corresponding to multimedia content, the paint stream data including semantic representation data and render instruction

data corresponding to the multimedia content, the multimedia content including play instructions and at least one multimedia element, the at least one multimedia element including at least one of graphical images, audio, text, and full motion video, the semantic representation data describing at least one of: characteristics of a rendered representation of the multimedia content, and relationships between different multimedia elements disposed in the rendered representation;

the computer rendering the data corresponding to the semantic representation and the data corresponding to the rendering instructions to generate a rendered representation of the multimedia content;

the computer storing the data corresponding to the semantic representation in a rendered cache coupled with the client; and

the computer storing the rendered representation in the rendered cache.

7. A method, implemented in at least one computer, the method, comprising:

detecting multimedia content including layout instructions, the multimedia content including at least one multimedia element, the at least one multimedia element including at least one of graphical images, audio, text, and full motion video;

laying out the multimedia content according to the layout instructions to form rendering instructions and a semantic representation for the multimedia content, the semantic representation describing at least one of: characteristics of a rendered representation of the multimedia content, and relationships between different multimedia elements disposed in the rendered representation;

storing data corresponding to the rendering instructions in a layout cache; and

storing data corresponding to the semantic representation in the layout cache.

8. The method of claim 7, wherein:

the layout cache is disposed at a client, the client is adapted to communicate with a server;

the server performs the detecting and generating steps; and

the method includes prior to storing the data corresponding to the rendering instructions:

the server transmitting data corresponding to the rendering instructions to the client; and

the server transmitting data corresponding to the semantic representation to the client.

9. The method of claim 8 comprising:

client processing resources rendering the multimedia content based on the rendering instructions and the semantic representation to form the rendered representation; and

playing the multimedia content based on the rendered representation.

10. The method of claim 7, wherein the layout cache is disposed at a server, and the server is adapted to communicate with a client.

11. The method of claim 10, including:

the client receiving the rendering instructions and the semantic representation;

client processing resources rendering the multimedia content based on the rendering instructions and the semantic representation to form the rendered representation;

the client formatting the rendered representation for rapid reading;

the client formatting the semantic representation into semantic content, the semantic content formatted for rapid reading and rapid restoration to the semantic representation; and

storing the rendered representation and the semantic content in a rendered cache coupled to the client.

12. The method of claim 10 comprising:

client processing resources rendering the multimedia content based on the rendering instructions and the semantic representation to form the rendered representation;

playing the multimedia content based on the rendered representation.

13. A method, implemented in a computer, for storing multimedia data, comprising:

the computer detecting paint stream data corresponding to multimedia content; the paint stream data includes semantic representation data and rendering instruction data corresponding to the multimedia content, the multimedia content including play instructions and at least one multimedia element, the at least one multimedia element including at least one of graphical images, audio, text, and full motion video, the semantic representation data describing at least one of: characteristics of a rendered representation of the multimedia content, and relationships between different multimedia elements disposed in the rendered representation;

the computer storing the semantic representation data in a layout cache coupled with the computer;

the computer storing the rendering instruction data in the layout cache.

14. A method, implemented in a server, for retrieving multimedia data, comprising:

server processing resources detecting a request for requested multimedia content;

server processing resources determining whether data corresponding to the requested multimedia content is disposed in a server cache coupled with the server, the server cache including rendered representations of multimedia content and semantic content, the semantic content including data corresponding to semantic representations derived from one of: play instructions for the rendered content, and layout of the multimedia content, the semantic representations describe at least one of: characteristics of the rendered

representations, and relationships between different multimedia elements disposed in the rendered representations; and

responding to a determination that data corresponding to the requested multimedia content are disposed in the server cache by:

retrieving a rendered representation of the requested multimedia content from the server cache; and

retrieving semantic content corresponding to the requested multimedia content from the server cache.

15. The method for retrieving multimedia data of claim 14, including, prior to retrieving the rendered representation of the requested multimedia content:

server processing resources determining whether the data corresponding to the requested multimedia content disposed in the server cache require updating:

responsive to a determination that the data corresponding to the requested multimedia content disposed in the server cache require updating:

storing an updated version of the data corresponding to the requested multimedia content in the paint stream cache, the updated version including updated data for at least a portion of the requested multimedia content and including data corresponding to rendering instructions and data corresponding to the semantic representation;

retrieving at least a portion of an updated version of the rendering instructions for the requested multimedia content from the paint stream cache;

retrieving at least a portion of an updated version of the semantic content corresponding to the requested multimedia content from the paint stream cache; and

restoring a semantic representation for the requested multimedia content using the at least a portion of the updated version of the semantic content.

16. The method for retrieving multimedia data of claim 14, including, prior to retrieving the rendered representation of the requested multimedia content:

processing resources coupled with the server determining whether the data corresponding to the requested multimedia content disposed in the server cache require updating; and

responsive to a determination that the data corresponding to the requested multimedia content disposed in the server cache require updating:

storing an updated version of the data corresponding to the requested multimedia content in the layout cache including an updated version of the rendering instructions for the requested multimedia content, and an updated version of the semantic content corresponding to the updated version of rendering instructions, wherein the updated version of the rendering instructions and the updated version of the semantic content include updated data for at least a portion of the requested multimedia content;

retrieving the updated version of the rendering instructions for the requested multimedia content;

retrieving the updated version of the semantic content corresponding to the updated version of the rendered representation; and

restoring the semantic representation for the requested multimedia content corresponding to the updated version of the rendering instructions using the updated version of the semantic content.

17. A cache comprising:

a storage medium; and

an indexing mechanism, the indexing mechanism adapted to store and retrieve:

rendering instructions for multimedia content formatted for rapid play, the multimedia content including at least one multimedia element, the at least one multimedia element includes at least one of graphical images, audio, text, and full motion video; and

semantic content of the multimedia content, the semantic content including data describing at least one of:

characteristics of a rendered representation of the multimedia content, the rendered representation corresponding to the rendering instructions; and

relationships between different multimedia elements disposed in the rendered representation.

18. A client comprising:

processing resources adapted to detect rendering instructions and semantic content of multimedia content, the multimedia content including at least one multimedia element, the at least one multimedia element including at least one of graphical images, audio, text, and full motion video, the semantic content including data describing at least one of: characteristics of a rendered representation of the multimedia content, and relationships between different multimedia elements disposed in the rendered representation;

processing resources adapted to respond to detecting the rendering instructions and the semantic content by forming a rendered representation of the multimedia content from the rendering instructions and the semantic content: and

processing resources adapted to play, according to the semantic content, at least a portion of a graphical representation of the multimedia content corresponding to the rendered representation.

19. The client of claim 18, including a set-top box having processing resources adapted to:

detect the rendering instructions and the semantic content of the rendered representations; and

play portions of the rendered representations;

wherein the semantic content includes data corresponding to scroll commands.

20. The client of claim 18, wherein the rendering instructions include multimedia elements corresponding to the multimedia content.

21. The client of claim 18, including:

processing resources adapted to, separate from a request for the rendering instructions, request multimedia elements corresponding to the multimedia content from a data source external to the client;

processing resources adapted to respond to at least one of the multimedia elements corresponding to the multimedia content not being included in the rendering instructions by forming a rendered representation of the multimedia content from the multimedia elements not included in the rendering instructions, the rendering instructions, and the semantic content.

22. The client of claim 18 including processing resources adapted to store the rendered representation and the semantic content in a storage medium coupled with the client.

23. A system for using multimedia content comprising:

web crawler processing resources adapted to access the multimedia content from source data storage, the multimedia content including at least one multimedia element, the at least one multimedia element including at least one of graphical images, audio, text, and full motion video;

layout processing resources adapted to:

generate rendering instructions for the multimedia content; and

generate a semantic representation of the multimedia content, the semantic representation describing at least one of: characteristics of a rendered representation of the multimedia content, and relationships between different multimedia elements disposed in the rendered representation of the multimedia content;

rendering processing resources adapted to:

format the semantic representation as semantic content; and

render the multimedia content into the rendered representation, the rendered representation formatted for rapid play; and

a cache including:

a storage medium; and

an indexing mechanism adapted to store and retrieve:

the rendering instructions of the multimedia content; and

the semantic content of the multimedia content

24. A system for accessing multimedia content comprising:

a cache including:

a storage medium; and

an indexing mechanism adapted to store and retrieve:

rendering instructions for generating a rendered representation of the multimedia content, the rendered representation is formatted for rapid play, the multimedia content including at least one multimedia element, the at least one multimedia element including at least one of graphical images, audio, text, and full motion video; and

semantic content of the multimedia content, the semantic content including data describing at least one of: characteristics of the rendered representation, and relationships between different multimedia elements disposed in the rendered representation;

layout processing resources adapted to:

lay out the multimedia content;

derive rendering instructions from a content definition;

generate a semantic representation of the multimedia content from lay out of the multimedia content; and

format the semantic representation as semantic content; and

rendering processing resources adapted to:

convert the multimedia content into the rendered representation; and

create a graphical representation of the multimedia content; and

wherein the rendering processing resources use the rendering instructions to create the graphical representation.

25. A method for playing multimedia content, comprising:

server processing resources retrieving rendering instructions of the multimedia content from a storage medium;

server processing resources retrieving semantic content of the multimedia content from the storage medium, the semantic content including data describing at least one of: characteristics of a rendered representation of the multimedia content corresponding to the rendering instructions, and relationships between different multimedia elements disposed in the rendered representation;

server processing resources transmitting:

an active portion of the rendering instructions, the active portion of the rendering instructions one of: a portion of the rendering instructions corresponding to a graphical representation presently being played, and a portion of the rendering instructions corresponding to a graphical representation to be played rapidly after transmitting; and

an active portion of the semantic content corresponding to the active portion of the rendering instructions;

client processing resources detecting the active portion of the rendering instructions and the active portion of the semantic content; and

client processing resources converting the rendering instructions and the semantic content into the rendered representation;

client processing resources reading the rendered representation and the semantic content;

client processing resources restoring the semantic representation of the graphical representation based on the semantic content; and

client processing resources playing the active portion of the graphical representation.

26. The method for playing multimedia content of claim 25, wherein:

the client includes a set-top box;

the multimedia content includes an image having at least one hyperlink;

the semantic representation includes at least one of:

a location of at least one hyperlink;

a size of at least one hyperlink;

a shape of at least one hyperlink; and

target index of at least one hyperlink.

27. A method for communicating and using multimedia data, comprising:

detecting multimedia content, the multimedia content includes play instructions and at least one multimedia element, the at least one multimedia element includes at least one of graphical images, audio, text, and full motion video;

generating a semantic representation and rendering instructions for the multimedia content from the play instructions, the play instructions include at least one of timing of the multimedia content and ordering of the multimedia content, the semantic representation describes at least one of: characteristics of a rendered representation of the multimedia content, and relationships between different multimedia elements disposed in the rendered representation;

storing data corresponding to the semantic representation in the paint stream cache on a client, the client adapted to communicate with a server; and

the client requesting multimedia content from the server based on the semantic representation in the paint stream cache.

28. A method involving multimedia data, the method comprising:

receiving multimedia content at a cable system headend;

detecting the multimedia content, the multimedia content including play instructions and at least one multimedia element, the at least one multimedia

element including at least one of graphical images, audio, text, and full motion video;

generating a semantic representation and rendering instructions for the multimedia content from the play instructions, the play instructions including at least one of timing of the multimedia content and ordering of the multimedia content, the semantic representation describing at least one of: characteristics of a rendered representation of the multimedia content, and relationships between different multimedia elements disposed in the rendered representation;

storing data corresponding to the rendering instructions in a paint stream cache; and

storing data corresponding to the semantic representation in the paint stream cache.

29. The method of claim 29, wherein:

the paint stream cache is disposed at a set top box at a user premises, the set top box is adapted to communicate with a server located at the cable system headend, and the set top box is coupled to a display;

the server performs the detecting and generating steps; and

the method includes prior to storing the data corresponding to the rendering instructions:

the server transmitting data corresponding to the rendering instructions to the set top box; and

the server transmitting data corresponding to the semantic representation to the set top box.

30. The method of claim 29, wherein:

the paint stream cache is disposed at a server located at the cable system headend, a set top box located at a user premises is adapted to communicate with the server, and the set top box is coupled to a display;

the server performs the detecting, generating, rendering, and playing; and

the set top box displays rendered content on the display in response to the rendering and playing by the server.

31. The method of claim 29, wherein:

the cable system headend receives the multimedia content from the Internet.

32. The method of claim 29, wherein the cable system headend receives the multimedia content via an HTTP connection

33. A method involving multimedia data, the method comprising:

receiving multimedia content at a server;

detecting the multimedia content, the multimedia content including play instructions and at least one multimedia element, the at least one multimedia element including at least one of graphical images, audio, text, and full motion video;

generating at a server semantic representation, rendering instructions, and partially rendered content from the play instructions, the play instructions including at least one of timing of the multimedia content and ordering of the multimedia content,

transmitting the semantic representation, the partially rendered content, and some of the rendering instructions to a client; and

generating at the client fully rendered multimedia content in response to the partial rendered content and the some of the rendering instructions.

34. The method of claim 33 including:

storing the semantic representation, the partially rendered content and the some of the rendering instructions in a cache at the client; and

after the storing, the client generating the fully rendered multimedia content in response to the partial rendered content and the some of the rendering instructions in the cache.

35. The method of claim 33, wherein semantic representation and the some of the rendering instructions are transmitted to the client before transmitting the partially rendered content, and wherein the partially rendered content is transmitted when the client requests the rendered content.

36. The method of claim 33, wherein the semantic representation, rendering instructions, and partially rendered content are stored at the server in a cache and are transmitted to the client when the client makes a request for the content.

37. The method of claim 33, wherein the partially rendered content corresponds to all non-text elements of the multimedia content.

38. The method of claim 33, wherein the server is located at a cable system headend and the client is located in a set top box at a user's residence.