

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AYLO FREESITES LTD,
Petitioner

v.

WELLCOMEMAT, LLC,
Patent Owner

IPR2024-00710
U.S. Patent No. 8,307,286

**DECLARATION OF HENRY HOUH, PH.D. IN
SUPPORT OF PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 8,307,286**

TABLE OF CONTENTS

CLAIM LISTING	vii
I. INTRODUCTION	1
II. BACKGROUND AND QUALIFICATIONS	1
III. MATERIALS CONSIDERED	10
IV. UNDERSTANDING OF LEGAL PRINCIPLES	12
A. Claim Construction	12
B. Anticipation	13
C. Obviousness.....	13
V. BACKGROUND OF THE TECHNOLOGY.....	17
A. Video Indexing	17
B. Video Streaming.....	19
C. Client-Server Systems	20
VI. OVERVIEW OF THE '286 PATENT	21
A. Summary of '286 Patent.....	21
B. Prosecution History	26
C. Priority Date	26
VII. LEVEL OF SKILL IN THE ART AND PERSPECTIVE APPLIED IN THIS DECLARATION	27
VIII. IDENTIFICATION OF CHALLENGES	28
IX. CLAIM CONSTRUCTION	28
X. GROUND 1: CLAIMS 13-15 AND 18-20 ARE ANTICIPATED BY GUPTA	29
A. Summary of Gupta	29
B. Claim 13.....	29
1. [13.P] A method for handling video cue points, the method comprising:.....	31
2. [13.1] providing a computer system having at least one computer-readable medium comprising a video editing	

	environment, and being disposed at a server side of a public network;	34
3.	[13.2] receiving a set of video information at the video editing environment using the computer system;	42
4.	[13.3] receiving a set of cue point information at the video editing environment using the computer system;	46
5.	[13.4] receiving a cue point editing command over the network from a remote client terminal disposed at a client side of the public network using the computer system;.....	52
6.	[13.5] modifying a portion of the cue point information at the server side of the public network according to the cue point editing command received over the public network from the client terminal;.....	62
7.	[13.6] receiving a cue point selection over the network from the remote client terminal identifying one of the cue points corresponding to a portion of the video information of one of the video files; and.....	65
8.	[13.7] streaming the portion of the video information to the client terminal over the public network in response to receiving the cue point selection.....	71
C.	Claim 14	75
D.	Claim 15	78
E.	Claim 18	81
F.	Claim 19	84
G.	Claim 20	85
XI.	GROUND 2: CLAIMS 13-15 AND 18-20 ARE ANTICIPATED BY FEGAN.....	88
A.	Summary of Fegan	88
B.	Claim 13.....	88
1.	[13.P] A method for handling video cue points, the method comprising:.....	90
2.	[13.1] providing a computer system having at least one computer-readable medium comprising a video editing	

	environment, and being disposed at a server side of a public network;	96
3.	[13.2] receiving a set of video information at the video editing environment using the computer system;	107
4.	[13.3] receiving a set of cue point information at the video editing environment using the computer system;	115
5.	[13.4] receiving a cue point editing command over the network from a remote client terminal disposed at a client side of the public network using the computer system;.....	122
6.	[13.5] modifying a portion of the cue point information at the server side of the public network according to the cue point editing command received over the public network from the client terminal;.....	129
7.	[13.6] receiving a cue point selection over the network from the remote client terminal identifying one of the cue points corresponding to a portion of the video information of one of the video files; and.....	130
8.	[13.7] streaming the portion of the video information to the client terminal over the public network in response to receiving the cue point selection.....	135
C.	Claim 14	138
D.	Claim 15	140
E.	Claim 18	143
F.	Claim 19	145
G.	Claim 20	147
XII.	GROUND 3: CLAIMS 13-15 AND 18-20 ARE OBVIOUS OVER KIM.....	149
A.	Summary of Kim	149
B.	Claim 13.....	149
1.	[13.P] A method for handling video cue points, the method comprising:.....	151
2.	[13.1] providing a computer system having at least one computer-readable medium comprising a video editing	

	environment, and being disposed at a server side of a public network;	155
3.	[13.2] receiving a set of video information at the video editing environment using the computer system;	164
4.	[13.3] receiving a set of cue point information at the video editing environment using the computer system;	169
5.	[13.4] receiving a cue point editing command over the network from a remote client terminal disposed at a client side of the public network using the computer system;.....	173
6.	[13.5] modifying a portion of the cue point information at the server side of the public network according to the cue point editing command received over the public network from the client terminal;.....	187
7.	[13.6] receiving a cue point selection over the network from the remote client terminal identifying one of the cue points corresponding to a portion of the video information of one of the video files; and.....	190
8.	[13.7] streaming the portion of the video information to the client terminal over the public network in response to receiving the cue point selection.....	197
C.	Claim 14	200
D.	Claim 15	206
E.	Claim 18	210
F.	Claim 19	212
G.	Claim 20	214
XIII.	GROUND 4: CLAIMS 13-15 AND 18-20 ARE OBVIOUS OVER KIM AND GUPTA.....	217
A.	Claim 13.....	217
1.	[13.4] receiving a cue point editing command over the network from a remote client terminal disposed at a client side of the public network using the computer system;.....	219
B.	Claim 14	220
C.	Claim 15	221
D.	Claims 18-20	222

XIV. GROUND 5: CLAIMS 13-15 AND 18-20 ARE OBVIOUS OVER VASILEVSKY AND LOGAN	223
A. Summary of Vasilevsky	223
B. Summary of Logan.....	223
C. Claim 13.....	224
1. [13.P] A method for handling video cue points, the method comprising:.....	225
2. [13.1] providing a computer system having at least one computer-readable medium comprising a video editing environment, and being disposed at a server side of a public network;	228
3. [13.2] receiving a set of video information at the video editing environment using the computer system;	242
4. [13.3] receiving a set of cue point information at the video editing environment using the computer system;	246
5. [13.4] receiving a cue point editing command over the network from a remote client terminal disposed at a client side of the public network using the computer system;.....	251
6. [13.5] modifying a portion of the cue point information at the server side of the public network according to the cue point editing command received over the public network from the client terminal;.....	257
7. [13.6] receiving a cue point selection over the network from the remote client terminal identifying one of the cue points corresponding to a portion of the video information of one of the video files; and.....	258
8. [13.7] streaming the portion of the video information to the client terminal over the public network in response to receiving the cue point selection.....	261
D. Claim 14	262
E. Claim 15	265
F. Claim 18	267
G. Claim 19	269
H. Claim 20	272

XV. CONCLUSION.....	274
---------------------	-----

CLAIM LISTING

The following claim listing adds bracketed labels for reference:

[13.P] 13. A method for handling video cue points, the method comprising:

[13.1] providing a computer system having at least one computer-readable medium comprising a video editing environment, and being disposed at a server side of a public network;

[13.2] receiving a set of video information at the video editing environment using the computer system;

[13.3] receiving a set of cue point information at the video editing environment using the computer system;

[13.4] receiving a cue point editing command over the network from a remote client terminal disposed at a client side of the public network using the computer system;

[13.5] modifying a portion of the cue point information at the server side of the public network according to the cue point editing command received over the public network from the client terminal;

[13.6] receiving a cue point selection over the network from the remote client terminal identifying one of the cue points corresponding to a portion of the video information of one of the video files; and

[13.7] streaming the portion of the video information to the client terminal over the public network in response to receiving the cue point selection.

14. The method of claim 13, further comprising providing a cue point editing mode as part of the video editing environment configured to perform at least one of creating, modifying, deleting, or interpreting at least a portion of the set of cue point information according to the cue point editing command received over the public network from the client terminal.

15. The method of claim 14, further comprising: receiving an edit mode entry signal over the network from the client terminal representing entry into the cue point edit mode, and switching the video editing environment to the cue point editing mode in response to receiving the edit mode entry signal.

18. The method of claim 13, further comprising receiving, over the network, authentication information relating to access authority of the remote client terminal.

19. The method of claim 18, further comprising denying or allowing access to all or part of the video editing environment based at least in part on the access authority of the remote client terminal.

20. The method of claim 13, wherein the server video data is streamed progressively to a video playback environment at the remote client terminal, allowing the video player to begin displaying streamed server video data while remaining server video data continues to be streamed.

I. INTRODUCTION

1. My name is Henry Houh, and I have been retained by counsel for Aylo Freesites, Ltd (“Petitioner”) to analyze U.S. Patent No. 8,307,286 (“’286 Patent”) (Exhibit 1001) and to provide my opinions regarding the patentability of claims 13-15 and 18-20 of the ’286 Patent.

2. I am being compensated at my normal consulting rate of \$650 per hour for my time. My compensation is not contingent on the outcome of this proceeding, or of any proceedings relating to the ’286 Patent.

3. This Declaration summarizes the opinions I have formed to date. I reserve the right to modify my opinions, if necessary, based on further review and analysis of information that I receive subsequent to the filing of this report, including in response to positions taken by WellcomeMat, LLC (“WellcomeMat”) or its experts that I have not yet seen, including any secondary consideration evidence that WellcomeMat or its expert may consider and present.

II. BACKGROUND AND QUALIFICATIONS

4. I am currently employed by Houh Consulting Inc. as an independent technical consultant. Until October 2022, I also worked as the co-founder of Einstein’s Workshop, a company that provides supplemental science, technology, engineering, and mathematics (“STEM”) education to children of all ages. I also co-

founded and serve on the board of BlocksCAD, Inc., which provides educational computer-aided design software used around the world.

5. My professional career has spanned more than 25 years. As set forth in my curriculum vitae (provided as Exhibit 1003), during these years I have gained extensive experience in web sites, communications systems, networking, distributed systems, protocols and streaming audio/video communications.

6. I have multiple undergraduate and graduate degrees from the Massachusetts Institute of Technology (MIT). I have a Bachelor of Science Degree in Electrical Engineering and Computer Science from MIT which I received in June 1989. I also have a Bachelor of Science Degree in Physics from MIT which I received in February 1990. I have a Master of Science Degree in Electrical Engineering and Computer Science from MIT which I received in February 1991. I have a PhD in Electrical Engineering and Computer Science from MIT which I received in February 1998. My thesis was directed to “Designing Networks for Tomorrow’s Traffic,” and was supervised by Professor David Tennenhouse and Professor John Guttag.

7. I first worked in the area of communications in 1987 when I worked at AT&T Bell Laboratories as part of a five-year dual degree program at MIT. At Bell Labs, I first worked in a group supporting core telephony switching functions, then later I joined the Optical Computing group. In graduate school, I joined the

Advanced Network Architecture (ANA) group at the Laboratory for Computer Science (LCS) at MIT, then headed by Prof. Tennenhouse and senior research scientist David Clark. Dr. Clark is generally considered to be one of the fathers of the Internet Protocol, having acted as chief protocol architect for the development of the Internet. My research and work experience in distributed systems, networking, and network architecture dates back to around 1991 when I joined ANA.

8. Later, when Prof. Tennenhouse formed his own research group, the Telemedia Network Systems (TNS) group, I joined TNS. The TNS group built a high-speed gigabit network and applications which ran over the network, such as remote video capture, processing, and display on computer terminals. I designed the core networking hardware and software, including the high-speed data links and the device drivers for the network interface cards.

9. TNS's research activities focused on high-speed networking and the types of applications that require increased network bandwidth and computing power. One of the applications I investigated was live streaming video, and real-time processing of such video. My work was focused on the transport, switching, and routing of data (including video and audio streams), and the integration of TNS's network into the computing environment, including the protocols, IP/ATM integration, operating system software and device drivers. TNS created a computing environment where it was possible to stream audio/video live, and also created

software processing modules to further process video to perform advanced functions such as green screening, scene change detection, motion detection, compression, video blending and video overlay, among many other features – all in real-time while displaying the processed video live. Our system also had the capability to identify a frame within video files, so that the video could be started at a particular point within the video. While the Internet at the time did not have the capability for such applications to be widely deployed, my group at TNS researched what would be possible when the required network and computing power became available. As discussed above, my Ph.D. thesis work focused on studying the routing resources required to route streams of video through the network, proposing several alternative labeling strategies to speed up network routing, especially for popular applications such as web traffic.

10. During the early part of my graduate studies, a time when there were perhaps a hundred or so web servers in existence, I set up a web server on one of our lab computers, and created content rich web pages for my research group and my research activities. Eventually, I, together with others I was working with, added live video demonstrations to TNS's web site. TNS's web site was one of the first several hundred web servers to exist, and the first to offer live video demonstrations initiated from the web site to a user's remote computer. I helped to build the web pages that initiated these video sessions via a web interface. I started and maintained the

“Commercial Sites Index,” the first listing of all the companies on the web. I later sold this list to OpenMarket, then a leader in web servers and eCommerce servers. I also started a company to build web sites and sell targeted online advertisements. During this time, I followed the evolution of the web very closely.

11. Vice-President Al Gore visited our group in 1996 and received a demonstration of – and remotely drove – a radio controlled toy car with a wireless video camera mounted on it; the video was encoded by TNS-designed hardware, streamed over the TNS-designed network and displayed using TNS-designed software.

12. I authored or co-authored at least twelve papers and conference presentations on my group’s research at TNS. I also co-edited TNS’s final report on its gigabit networking research effort. Some of my papers about web site development were presented at the first World-Wide Web conference held in 1994 at CERN in Geneva, Switzerland. CERN is the birthplace of the Web, and the conference was chaired by Tim Berners-Lee, the inventor of the World-Wide Web. When Tim Berners-Lee formed the World Wide Web Consortium at the MIT LCS in order to manage the progression of web protocols and standards, I assisted him in obtaining the consortium’s domain names and setting up the consortium’s web server.

13. I have extensive experience in creating web pages and commercially setting up websites during the 1993-1997 time frame, and have continued this development since then. As part of this experience, I regularly programmed using HTML and Perl programming languages. I also have experience with CGI scripts and Perl scripts as used for dynamic web pages.

14. From 1997 to 1999, I was a Senior Scientist and Engineer at NBX Corporation, a start-up that made business telephone systems for streaming packetized audio over data networks instead of using traditional telephone lines. NBX was later acquired by 3Com Corporation, and the phone system I worked on at NBX was widely used by numerous businesses. As part of my work at NBX, I designed the core audio reconstruction algorithms for the telephones, as well as the packet transmission algorithms. I also designed and validated the core packet transport protocol used by the phone system.

15. Two of the company founders and I obtained U.S. Patent No. 6,967,963, entitled “Telecommunication method for ensuring on-time delivery of packets containing time-sensitive data,” as a result of part of the work I did at NBX.

16. From 2003-2004, I also was the architect for a next-generation web testing platform developed by Empirix, known as e-Test Suite. E-Test Suite is now owned by Oracle Corporation. E-Test provided functional and load testing for web sites. E-Test emulated a user’s interaction with a web site and provided web

developers with a method of creating various scripts and providing both functional testing (i.e., did the web site provide the correct response) and load testing (i.e., could the web site handle 5000 users simultaneously). As part of this project, I co-wrote a javascript interpreter for e-Test so that it could understand the javascript code present on advanced web sites.

17. I worked at BBN Technologies from 2004 to 2009. BBN Technologies, formerly known as Bolt, Beranek and Newman, was a pioneer of the Internet and a long-established leader in speech recognition and natural language processing. BBN received the first contract to build components of a packet switched network as part of the ARPANET, a U.S. Department of Defense effort funded through the Advanced Projects Research Agency (ARPA). While this effort preceded my time at BBN by many years, BBN continued to maintain a strong presence in network research and development, including in wireless networking, device discovery, and protocols for a “body area network.”

18. In 2006, as part of my role at BBN Technologies, I helped found PodZinger Inc., now known as RAMP Inc. PodZinger utilized BBN’s speech recognition algorithms to search through the spoken words inside audio and video. After managing the creation of the initial prototype system, PodZinger built out a full web-based streaming audio and video search solution when I was the Vice President of Operations and Technology there. The web site had a full load- balanced

back end that provided search and indexing solutions for our hosted multimedia (audio and video) content. Our PodZinger system processed videos in order to perform automatic speech recognition on the spoken contents of the video, after which we produced a word transcript we used in a search index for all our processed videos. All of the words in the transcript were individually tagged to frame locations within the video, so that selecting a word in the transcript caused the video to jump to the location within the video at which the word was spoken. For this work and more, we received a patent, on which I was an inventor – U.S. Patent No. 7,801,910, titled “Method and apparatus for timed tagging of media content.” In addition, we received several more patents on which I was an inventor, including U.S. Patent No. 9,697,231, titled “Methods and apparatus for providing virtual media channels based on media search,” and U.S. Patent No. 9,697,230, titled “Methods and apparatus for dynamic presentation of advertising, factual, and informational content using enhanced metadata in search-driven media applications.” These patents described computerized method and apparatus for merging video content segments from discrete media content for playback, and using metadata extracted from multimedia content segments to display additional information, such as advertisements, adjacent to the media player, while tracking the progression of the playback and changing the information as playback progressed through various segments to correspond to the content of the segments.

19. I also worked on commercializing a service offered by BBN to monitor call centers and provide analytical data on each call. The business became known as AVOKE Analytics and provided call center analytics for large organizations such as Dell and the Centers for Medicare and Medicaid Services (CMMS). For a time, during the rollout of the Affordable Care Act (though I was no longer working at BBN), AVOKE Analytics gathered data and provided call-by-call analytics for every call made to the CMMS main call center.

20. I have web advertising experience as well, including starting a company in 1994 that was funded, in part, through web advertisements. Later in 2008-2009, as the CTO of a venture-funded social network start-up company, I also evaluated different advertising network tools which were used in the roll-out of an advertising network.

21. I am familiar with patents and have numerous provisional and non-provisional patent applications on which I am an inventor, including 4 provisional patent applications, and around 20 non-provisional applications. These patent applications deal with various areas of technology including, but not limited to: utilizing network processors, network and packet delivery systems, tagging media content, dynamic presentation of advertising and other content using enhanced metadata, methods for web site testing, and organizing and managing files that are transmitted over a network.

22. Additional details about my employment history, fields of expertise, courses taught, and publications are further included in my CV attached as Appendix A to this Report.

III. MATERIALS CONSIDERED

23. In preparing this Declaration, I reviewed the '286 Patent, including the claims of the patent in view of the specification, and I have reviewed the prosecution history of the '286 Patent and numerous prior art and technical references from the time of the alleged invention. Of the materials cited as an exhibit to the '286 Patent IPR petition, I reviewed the following:

Exhibit	Description
1001	U.S. Patent No. 8,307,286 (“’286 patent”)
1004	U.S. Patent Application Publication No. 2005/0081159 (“Gupta”)
1005	U.S. Patent Application Publication No. 2005/0144305 (“Fegan”)
1006	U.S. Patent Application Publication No. 2005/0210145 (“Kim”)
1007	U.S. Patent Application Publication No. 2002/0069218 (“Sull”)
1008	WO 2003/067594 A1 (“Vasilevsky”)
1009	U.S. Patent Application Publication No. 2003/0093790 (“Logan”)
1010	File wrapper for '286 Patent

1011	Ulead DVD MovieFactor 2 User Guide (“Ulead”)
1012	U.S. Patent Application Publication No. 2004/0230600 (“Devore”)
1013	U.S. Patent Application Publication No. 2003/0231198 (“Janevski”)
1014	U.S. Patent No. 6,637,031 (“Chou”)
1015	U.S. Patent Application Publication No. 2002/0065879 (“Ambrose”)
1016	U.S. Patent Application Publication No. 2001/0020255 (“Hofmann”)
1017	Excerpts from Microsoft Computer Dictionary, 5th Ed. (2002)
1018	“Signal.” Merriam-Webster.com Dictionary, Merriam-Webster, https://www.merriam-webster.com/dictionary/signal (accessed March 29, 2024)

24. Petitioner’s counsel has asked me to consider whether certain references disclose or suggest, alone or in combination, the features recited in certain claims of the ’286 Patent. I have also been asked to consider the state of the art and the prior art available before the time of the alleged invention of the ’286 Patent. My opinions are provided in this Declaration.

25. My opinions in this Declaration are based on my review of the documents above, my understanding as an expert in the relevant field, and my education, training, research, knowledge, and personal and professional experience.

26. To my knowledge, I have no financial interest in Petitioner. Counsel for Petitioner has informed me that WellcomeMat purports to own the '286 Patent. To the best of my knowledge, I have no financial interest in WellcomeMat and, to my recollection, have had no contact with WellcomeMat or the named inventors of the '286 Patent. To the best of my knowledge, I do not have any financial interest in the '286 Patent.

27. To the extent any mutual funds or other investments that I own have a financial interest in the Petitioner, the Patent Owner, or the '286 Patent, I am not aware of, and do not control, any financial interest that would affect or bias my judgment.

IV. UNDERSTANDING OF LEGAL PRINCIPLES

A. Claim Construction

28. I understand that claim construction in an *Inter Partes* Review is controlled by the standard in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc).

29. I have been informed that claim construction begins with the language of the claims. The words are given their “ordinary and customary meaning,” which is “the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention, i.e., as of the effective filing date of the patent application.” The specification and prosecution history are sources of intrinsic

evidence that informs the meaning of the claim language. Extrinsic evidence, such as expert testimony and dictionaries, may be useful in educating judges regarding the field of the invention or helping determine what a person of ordinary skill in the art (“POSA”) would understand claim terms to mean. However, extrinsic evidence in general is viewed as less reliable than intrinsic evidence.

30. I understand that in this proceeding, any claim term that is not construed should be given its plain and ordinary meaning.

B. Anticipation

31. I understand that a claim is unpatentable as anticipated if all of the elements of the claim are described or disclosed in a single prior art reference.

C. Obviousness

32. I have been informed by counsel and understand that a prior art reference can render a patent claim obvious to one of ordinary skill in the art if the differences between the subject matter set forth in the patent claim and the prior art are such that the subject matter of the claim would have been obvious at the time the alleged invention was made. In analyzing obviousness, I understand that it is important to consider the scope of the claims, the level of skill in the relevant art, the scope and content of the prior art, the differences between the prior art and the claims, and any secondary considerations.

33. I have been informed by counsel and understand that a single reference can render a patent claim obvious if any differences between that reference and the claims would have been obvious to a person of ordinary skill in the art. Alternatively, the teachings of two or more references may be combined in the same way as disclosed in the claims, if such a combination would have been obvious to one having ordinary skill in the art. In determining whether a combination based on either a single reference or multiple references would have been obvious, I have been informed by counsel and understand that it is appropriate to consider the following factors:

- Whether the teachings of the prior art references disclose known concepts combined in familiar ways, and when combined, would yield predictable results;
- Whether a POSITA could implement a predictable variation, and would see the benefit of doing so;
- Whether the claimed elements represent one of a limited number of known design choices, and would have a reasonable expectation of success by those skilled in the art;
- Whether a person of ordinary skill would have recognized a reason to combine known elements in the manner described in the claim;

- Whether there is some teaching or suggestion in the prior art to make the modification or combination of elements claimed in the patent; and
- Whether the innovation applies a known technique that had been used to improve a similar device or method in a similar way.

34. I been informed by counsel and understand that a POSITA has ordinary creativity and is not an automaton.

35. I been informed by counsel and understand that all prior art references are to be looked at from the viewpoint of a POSITA.

36. I been informed by counsel and understand that, in considering obviousness, it is important not to determine obviousness using the benefit of hindsight derived from the patent being considered, and that obviousness is analyzed from the perspective of a POSITA at the time of the invention.

37. I been informed by counsel and understand that when the claimed subject matter involves combining pre-existing elements to yield no more than one would expect from such an arrangement, the combination is obvious. I also I been informed by counsel and understand that in assessing whether the claimed subject matter is obvious, one must consider whether the claimed improvement is more than the predictable use of prior art elements according to their established functions. I

been informed by counsel and understand that there need not be a precise teaching in the prior art directed to the specific subject matter of a claim because one can take account of the inferences and creative steps that a person of skill in the art would employ.

38. I been informed by counsel and understand that obviousness cannot be based on the hindsight combination of components selectively culled from the prior art. I been informed by counsel and understand that in an obviousness analysis, neither the motivation nor the avowed purpose of the inventors controls the inquiry. Any need or problem known in the field at the time of the invention and addressed by the patent can provide a reason for combining elements. For example, I been informed by counsel and understand that it is important to consider whether there existed at the time of the invention a known problem for which there was an obvious solution encompassed by the patent's claims. I been informed by counsel and understand that known techniques can have obvious uses beyond their primary purposes, and that in many cases a person of ordinary skill can fit the teachings of multiple pieces of prior art together like pieces of a puzzle.

39. I been informed by counsel and understand that, when there is a reason to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp. I further have been informed by counsel and understand that, if

this leads to the anticipated success, it is likely the product not of innovation but of ordinary skill and common sense, which bears on whether the claim would have been obvious.

40. I been informed by counsel and understand that secondary considerations can include, for example, evidence of commercial success of the invention, evidence of a long-felt need that was solved by an invention, evidence that others copied an invention, or evidence that an invention achieved a surprising result. I further have been informed by counsel and understand that such evidence must have a nexus or causal relationship to the elements of a claim, in order to be relevant. I am unaware of any such secondary considerations for the '286 Patent.

V. BACKGROUND OF THE TECHNOLOGY

A. Video Indexing

41. By 2002, DVD editing/burning software such as Ulead DVD MovieFactory was available to ordinary consumers, and such software allowed users to create custom DVDs from home movies using software on their personal computer. Ulead, pp. 1-2 (2002 dates), 4. This software also allowed users to “tag” movie segments at the point in the video the segment began. A user would provide such a tag by adding chapter information (similar to commercial DVDs) and a “text description” for each chapter. *Id.*, pp. 12-13. Viewers of the DVD could use this

chapter information to more easily navigate to a desired portion of the DVD movie.

Id., p. 12.

42. By 2003, it was well-known for users to use computer software to create, delete, and edit bookmarks (also referred to as videomarks) corresponding to certain times in a video such that the bookmarks/videomarks can be displayed to allow a viewer to navigate to the desired bookmark/videomark without needing to watch the entire video. *See, e.g.*, Janevski, Abstract, ¶¶[0009] (“videomarks may be displayed in a videomark menu or in a videomark timeline in which a videomark display icon represents each videomark”), [0046], [0058], [0063].

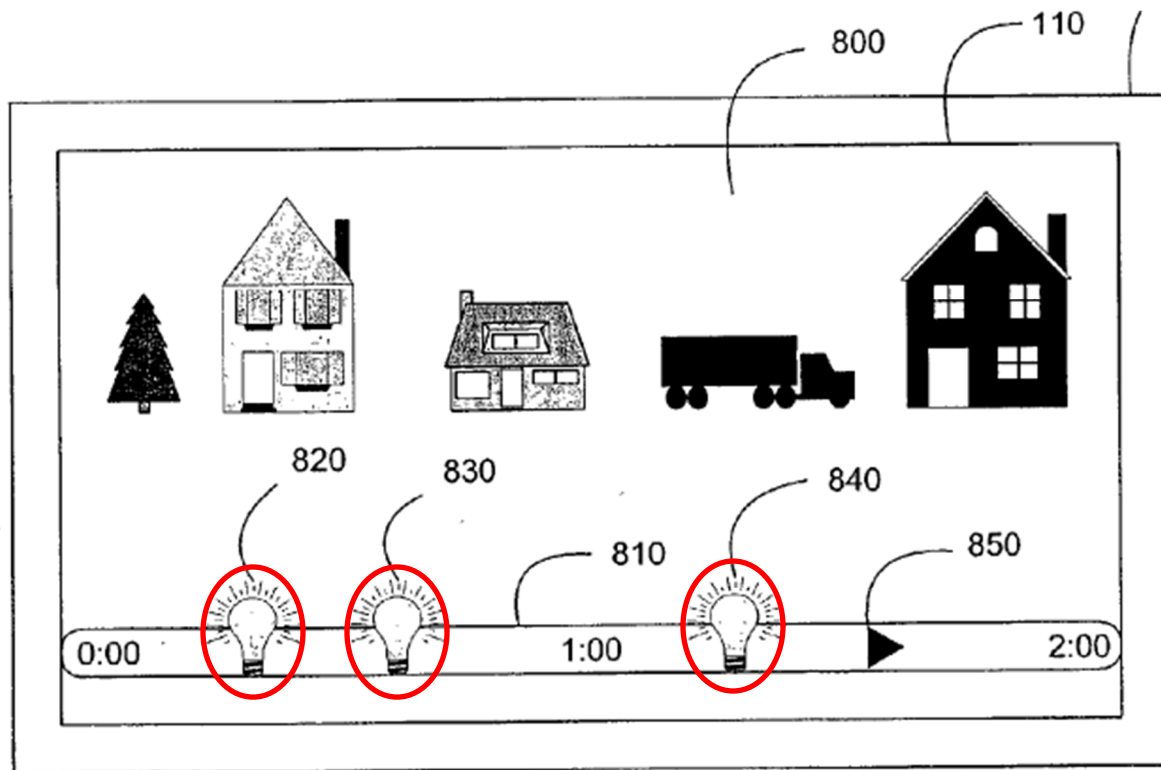


FIGURE 8

Id., FIG. 8 (annotated, showing videomarks saved to a video timeline).

43. The '286 Patent simply extends these known chaptering concepts from DVD videos to online video using a typical client-server framework, in which the editing environment is installed on a server (as opposed to a personal computer) and allowing users to connect to the server from their personal computers to add/edit/delete chapters of online videos.

B. Video Streaming

44. By 2003, technology for streaming over the Internet was becoming “increasingly important with the growth of the internet.” Chou, 2:13-14. A typical

streaming network architecture included video capturing tools to capture and store video on a video server which is connected to video clients over a network, such as the Internet, to progressively stream video from the server to the clients. *Id.*, FIG. 2, 6:34-64, 7:13-20.

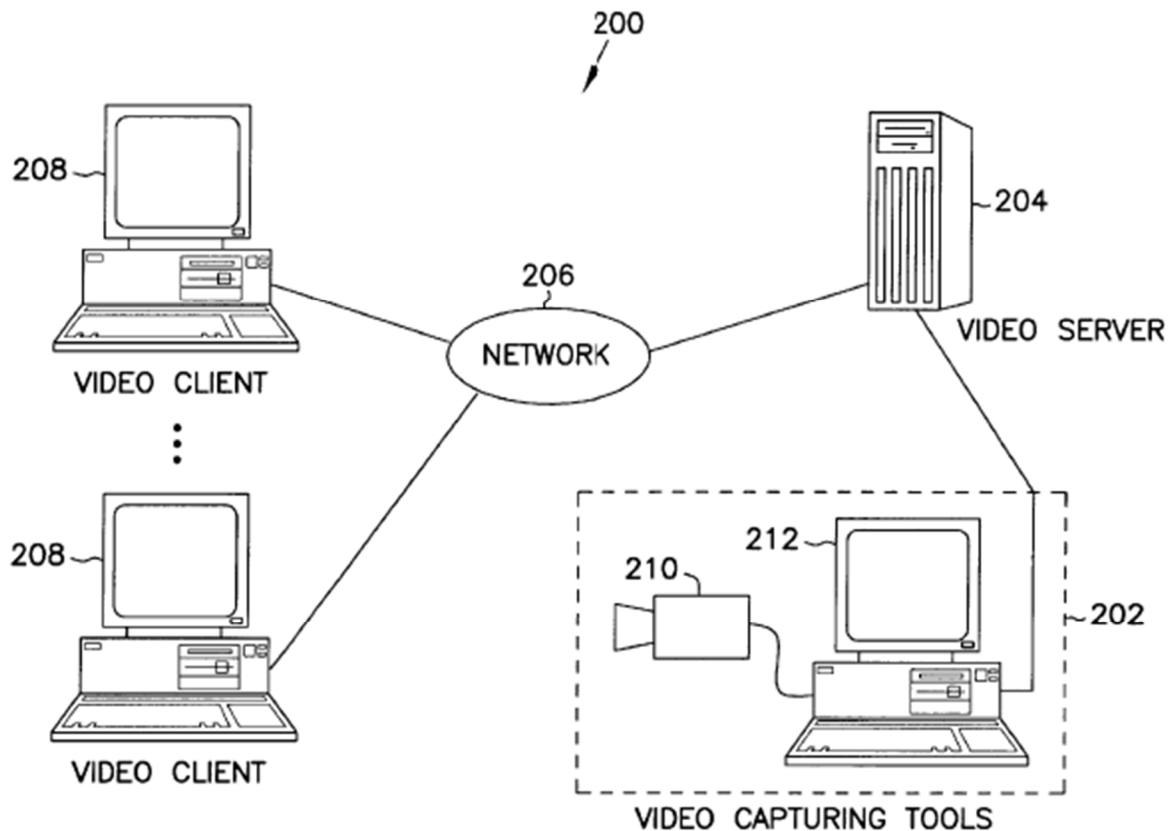


FIG. 2

Id., FIG. 2.

C. Client-Server Systems

45. Also by the early 2000s, it was well-known to implement systems according to a client-server architecture with clients connected to servers over the

public World Wide Web (Internet) such that the client can access and use software stored and executed at the server without needing to install and execute the software locally. *See, e.g.*, Ambrose, Abstract, ¶¶[0002]-[0014]; Hofmann, Abstract, [0004]-[0017].

VI. OVERVIEW OF THE '286 PATENT

A. Summary of '286 Patent

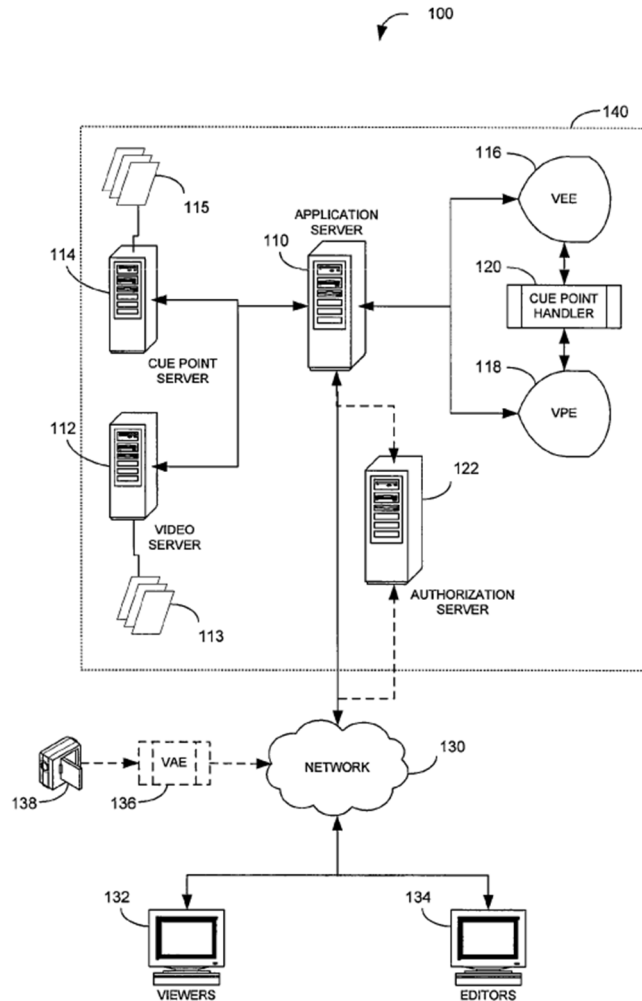
46. The '286 Patent relates to adding “cue points,” which may be “indexed time positions or tags,” to videos. '286 Patent, Abstract, 7:50-52. In an exemplary video of a hotel suite, cue points for “the bedroom, bathroom, living room, and kitchenette” are defined at “time positions” “zero, one, two, and three minutes” of the video. *Id.*, 7:49-60. As in chaptered DVDs, cue points “provid[e] chapter markers for the different sections of the video.” *Id.*

47. FIG. 1A shows a cue point handling system with application server 110 executing a Video Editing Environment (“VEE”) 116 and/or a Video Playback Environment (“VPE”) 118; video server 112 storing videos; and cue point servers 114 storing and serving cue point information. *Id.* 6:61-7:8, 7:42-45, 9:19-46, 10:4-9. These servers are connected to each other and to clients (*e.g.*, viewer terminals 132 interacting with the VPE; editor terminals 134 interacting with the VEE) via network 130 (*e.g.*, the Internet). *Id.* The servers’ functions may be either distributed among multiple servers (that are not necessarily collocated) or contained in one

server. *Id.*, 9:1-17. The VEE and VPE may be the same application or separate applications. *Id.*, 9:18-28.

48. A client may access the VEE via a GUI provided “as a webpage on a client’s web browser” (*id.*, 13:39-67) or as a “server-level application” (*id.* 12:63-13:6). The VEE is described broadly and “may comprise **any** graphical, textual, audio, or other data which is desirable or necessary **for providing interface capabilities to a client**” and “does not [even] require any graphical interface elements” (*e.g.*, “script”-level interfaces). *Id.*, 13:48-67 (emphasis added).

49. Authorization server 122 “allow[s] or den[ies] access to all or part of the various applications handled by the application server 110.” *Id.*, 10:20-26.



Id., FIG. 1A.

50. A client may load a cue point editor in the VEE to “add[], modify[], or delet[e] cue point information.” *Id.*, 19:6-35. FIG. 6 depicts an exemplary VEE cue point editor screen 600 displayed on a client. *Id.*, 25:64-26:60.

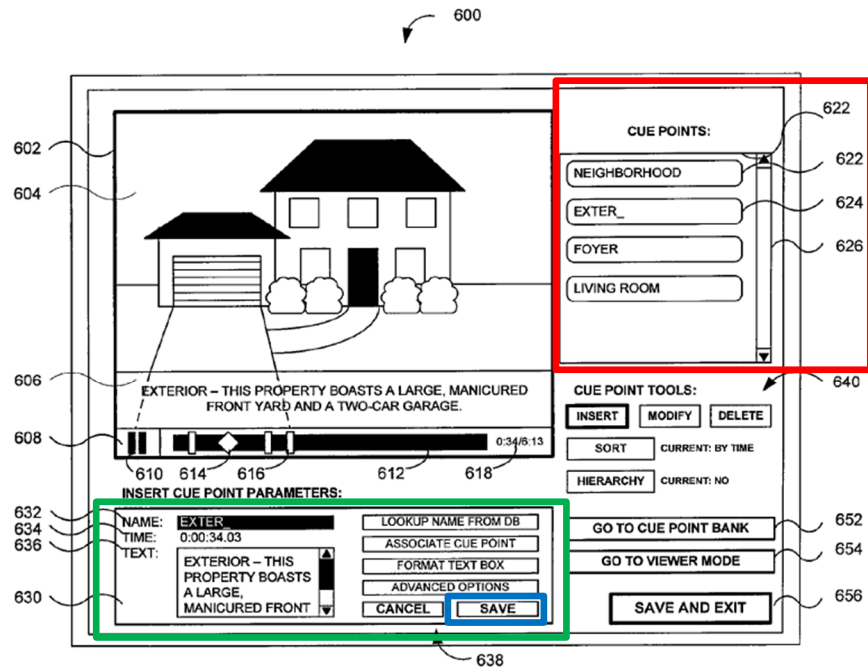


Fig. 6

Id., FIG. 6 (annotated).

51. The cue point editor screen 600 includes:

- Cue point listing area 620 (annotated above in red; misnumbered as the upper instance of “622”) listing cue points (*e.g.*, existing cue point 622 (“NEIGHBORHOOD”) and cue point 624 (“EXTER_”) that is being inserted). *Id.*, 27:21-31;
- Cue point parameters area 630 (annotated above in green) for inputting cue point information, such as cue point “name,” “time position,” and “comment text.” *Id.*, 27:32-40; and

- “Save” button (annotated above in blue) that “allow[s] a client to save the current cue point editing and possibly submit the cue point information.” *Id.*, 27:40-59, 23:24-28.

52. FIG. 11A depicts the VPE where a client accesses video and cue point information (*id.*, 34:59-35:8), including “cue point control area 1120” and “cue point controls 1122.” The client clicks on a cue point control (*e.g.*, 1122-1 through 3) “to navigate to those corresponding locations in the video.” *Id.*, 36:9-21.

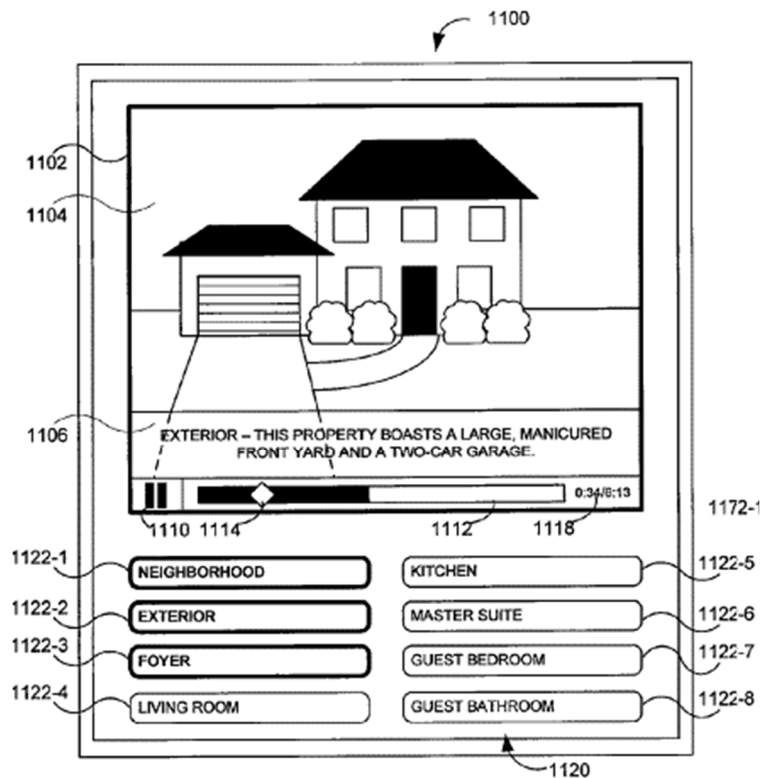


Fig. 11A

53. *Id.*, Fig. 11A.

B. Prosecution History

54. The underlying prosecution involved multiple rounds of Office Actions and applicant responses.

55. Ultimately, the Examiner allowed the application, and in the Notice of Allowance, provided reasons for allowance stating:

The prior art of record does not teach or fairly suggest the limitations “receive a cue point selection over the public network from the client terminal identifying one of the cue points corresponding to a portion of the video information of one of the video files; **and stream the portion of the video information** to the client terminal over the public network in response to receiving the cue point selection **and without storing the video file in its entirety to the client terminal.**”

Exhibit 1010, p. 16 (emphasis added).

56. However, while the “stream the portion of the video information ... without storing the video file in its entirety to the client terminal” language is present in independent claim 1, independent claim 13 does not recite any such limitation of performing streaming “without storing the video file in its entirety to the client terminal.”

C. Priority Date

57. I understand that the '286 Patent purports to claim the benefit of a provisional application filed on May 7, 2006. As the prior art dates of all references

in the unpatentability grounds are prior to May 7, 2006, I have not analyzed whether the provisional application provides written description support for any claim of the '286 Patent.

58. I reserve the right to supplement my opinions to address the proper effective date of the '286 Patent claims, should WellcomeMat allege an effective date earlier than May 7, 2006 during the IPR proceeding.

VII. LEVEL OF SKILL IN THE ART AND PERSPECTIVE APPLIED IN THIS DECLARATION

59. In my opinion, a person having ordinary skill in the art ("POSITA") as of the Priority Date would have had at least a Bachelor's degree in Electrical Engineering, Computer Engineering, or Computer Science (or the equivalent degree) plus approximately two years of work experience with technologies relating to video annotation/bookmarking and/or video navigation. A higher level of education or specific skill might make up for less experience, and vice-versa. I was a POSITA at the time of the Priority Date.

60. My opinions in this Declaration are based on the perspective of a POSITA as of the relevant timeframes discussed above, or somewhat before (approximately 2006-2007). This is true even if the testimony is stated in the present tense. Each of the statements below reflects my opinion based on my review of the documents listed in the table above.

VIII. IDENTIFICATION OF CHALLENGES

Ground 1: Claims 13-15 and 18-20 are anticipated by U.S. Patent Publication No. 2005/0081159 (“Gupta”; Exhibit 1004).

Ground 2: Claims 13-15 and 18-20 are anticipated by U.S. Patent Publication No. 2005/0144305 (“Fegan”, Exhibit 1005).

Ground 3: Claims 13-15 and 18-20 are obvious over U.S. Patent Publication No. 2005/0210145 (“Kim”; Exhibit 1006).

Ground 4: Claims 13-15 and 18-20 are obvious over Kim and Gupta.

Ground 5: Claims 13-15 and 18-20 are obvious over WIPO Publication No. 03/067594 A1 (“Vasilevsky”; Exhibit 1008).

IX. CLAIM CONSTRUCTION

61. I have generally applied the plain and ordinary meaning of the terms recited in the claims, as would have been understood by persons of skill in the art in the 2006-2007 timeframe.

62. I am not currently aware of any particular claim term whose specific construction would alter my opinions herein. Nonetheless, I reserve the right to respond to any claim constructions advanced by Patent Owner.

X. GROUND 1: CLAIMS 13-15 AND 18-20 ARE ANTICIPATED BY GUPTA

A. Summary of Gupta

63. Gupta (Exhibit 1004) is a publication of a U.S. patent application, published on April 14, 2005. I understand that Gupta constitutes prior art at least under pre-AIA 35 U.S.C. §§ 102(a), (b), and (e) even under the alleged May 7, 2006 priority date.

64. Gupta describes methods and systems in a client-server architecture for transmitting “media content as well as annotations corresponding to the media content ... from a server(s) to a client.” Gupta, Abstract. Gupta further describes “creating, viewing, and temporarily positioning annotations for media content.” *Id.*, ¶¶[0003]; *see also* ¶¶[0006]-[0007], [0010]. Gupta describes a user editing the content of an annotation (*id.*, ¶¶[0124]-[0128]) and a user selecting an annotation to seek/navigate to the temporal position of the media content corresponding to the annotation. *Id.*, ¶¶[0007], [0053], [0135]-[0136].

B. Claim 13

65. Claim 13 recites:

[13.P] 13. A method for handling video cue points, the method comprising:

[13.1] providing a computer system having at least one computer-readable medium comprising a video editing

environment, and being disposed at a server side of a public network;

[13.2] receiving a set of video information at the video editing environment using the computer system;

[13.3] receiving a set of cue point information at the video editing environment using the computer system;

[13.4] receiving a cue point editing command over the network from a remote client terminal disposed at a client side of the public network using the computer system;

[13.5] modifying a portion of the cue point information at the server side of the public network according to the cue point editing command received over the public network from the client terminal;

[13.6] receiving a cue point selection over the network from the remote client terminal identifying one of the cue points corresponding to a portion of the video information of one of the video files; and

[13.7] streaming the portion of the video information to the client terminal over the public network in response to receiving the cue point selection.

66. In my opinion, Claim 13 is anticipated by Gupta.

1. [13.P] A method for handling video cue points, the method comprising:

67. In my opinion, to the extent the preamble is limiting, Gupta discloses limitation [13.P]. Gupta discloses a method for handling (*e.g.*, “creating, viewing, and temporally positioning”) video cue points (*e.g.*, “annotations”) for media content such as videos.

This invention relates to networked client/server systems and to **annotating media content** in such systems. More particularly, the invention relates to **creating, viewing, and temporally positioning annotations** for media content.

Gupta, ¶[0003] (emphasis added).

The advent of computers and their continued technological advancement has revolutionized the manner in which people work and live. An example of such is in the education field, wherein educational presentations (*e.g.*, college lectures, workplace training sessions, etc.) can be provided to a computer user as **multimedia data (e.g., video,** audio, text, and/or animation data). Today, such presentations are primarily video and audio, but a richer, broader digital media era is emerging. Educational multimedia presentations provide many benefits, such as allowing the presentation data to be created at a single time yet be presented to different users at different times and in different locations throughout the world.

Gupta, ¶[0004] (emphasis added).

Streaming media server computer 11 has access to **streaming media content** in the form of different media streams. These media streams can be individual media streams (e.g., audio, **video**, graphical, etc.), or alternatively composite media streams including two or more of such individual streams. ...

Gupta, ¶[0030] (emphasis added).

68. Gupta's methods are implemented on "a networked client/server system" for videos, in which a client user uses a user interface to create annotations corresponding to videos. As discussed later, these annotations are transmitted from the client to a server.

In a **networked client/server system**, **media content** as well as **annotations** corresponding to the media content can be **transmitted from a server(s) to a client**. A **user interface** is presented to the user to facilitate **creating new annotations and viewing annotations**. According to one embodiment, the client further assists the user in identifying a temporal range of the media content to which the new annotation is to correspond.

Id., Abstract (emphasis added).

69. Gupta's annotation, created by a user via the user interface, is data (such as audio, text, video) that corresponds to a particular point in time in a video.

Multimedia presentations may also include **"annotations relating to the multimedia presentation"**. An annotation is data (e.g., **audio, text, video**, etc.) that corresponds to a multimedia

presentation. Annotations can be added by anyone with appropriate access rights to the annotation system (e.g., the lecturer/trainer or any of the students/trainees). These **annotations** typically **correspond to a particular temporal location in the multimedia presentation** and can provide a replacement for much of the "in-person" interaction and "classroom discussion" that is lost when the presentation is not made "in-person" or "live". As part of an annotation, a student can comment on a particular point, to which another student (or lecturer) can respond in a subsequent annotation. This process can continue, allowing a "classroom discussion" to occur via these annotations.

Id., ¶[0007] (emphasis added); *see also* ¶[0006].

According to one aspect of the invention, **annotations correspond to a temporal range of the media content** as defined by a **temporal beginning point and a temporal ending point**. ...

Id., ¶[0010] (emphasis added).

70. Similarly, the '286 Patent describes "cue points" as indexed time positions or tags anywhere within a video file. The '286 Patent also describes that that "cue points" may include "meta-tags" that describe the information in the video and include descriptive identifiers.

In some embodiments, cue points may refer to **indexed time positions or tags anywhere within a video file**.

'286 Patent, 7:50-52 (emphasis added).

In yet other embodiments, cue points may be video meta-tags. Meta-tags may describe the information in the video file, rather than pointing to specific data locations within the file. For example, video meta-tags for a video about a hotel Suite in downtown Denver may include descriptive identifiers, like “hotel,” “suite,” “kitchenette,” “luxury,” “downtown,” etc.

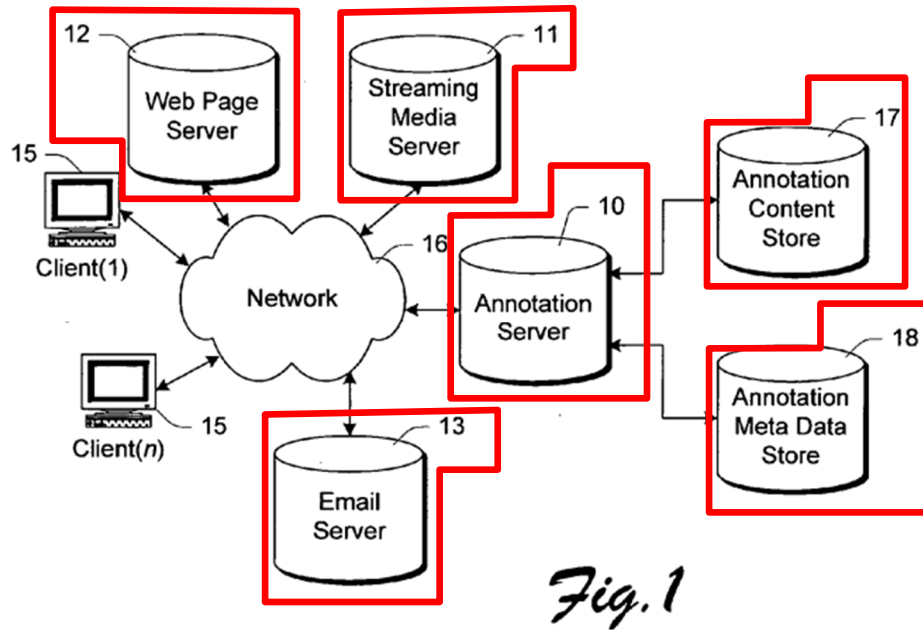
Id., 8:15-20 (emphasis added).

71. Therefore, Gupta’s annotations are “cue points” in the context of the '286 Patent claims.

2. **[13.1] providing a computer system having at least one computer-readable medium comprising a video editing environment, and being disposed at a server side of a public network;**

72. In my opinion, Gupta discloses limitation [13.1].

73. Gupta discloses providing a computer system that includes various servers and storage devices at a server side of a public network, including an annotation server 10, a streaming media server 11, a web page server 12, an email server 13, and stores 17 and 18. Gupta, ¶¶[0032]-[0034].

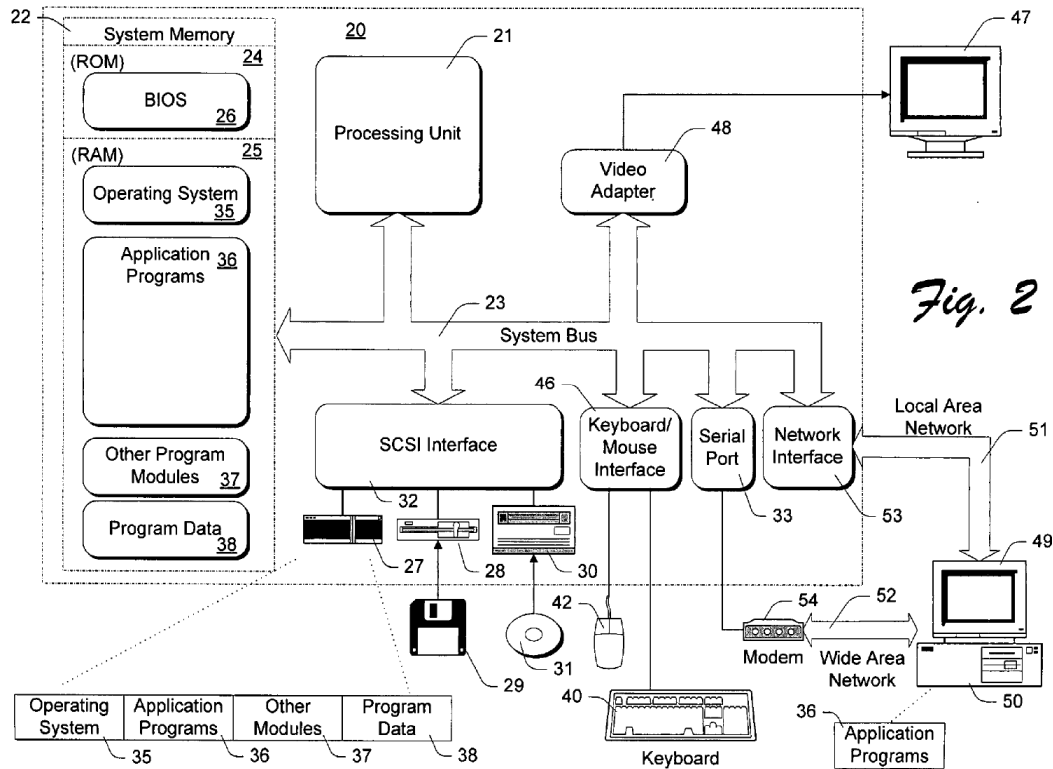


Gupta, FIG. 1 (annotated); *see also* ¶¶[0032]-[0034].

74. Gupta discloses that each of these servers has at least one computer-readable medium (*e.g.*, hard drives, RAM, ROM). In FIG. 2, Gupta illustrates an exemplary computer 20 that may be used to implement any of the servers.

FIG. 2 shows a general example of a computer 20 that can be used as a client or server in accordance with the invention. Computer 20 is shown as an example of a computer that can perform the functions of any of server computers 10-13 or a client computer 15 of FIG. 1.

Gupta, ¶[0037] (emphasis added).



Id., FIG. 2.

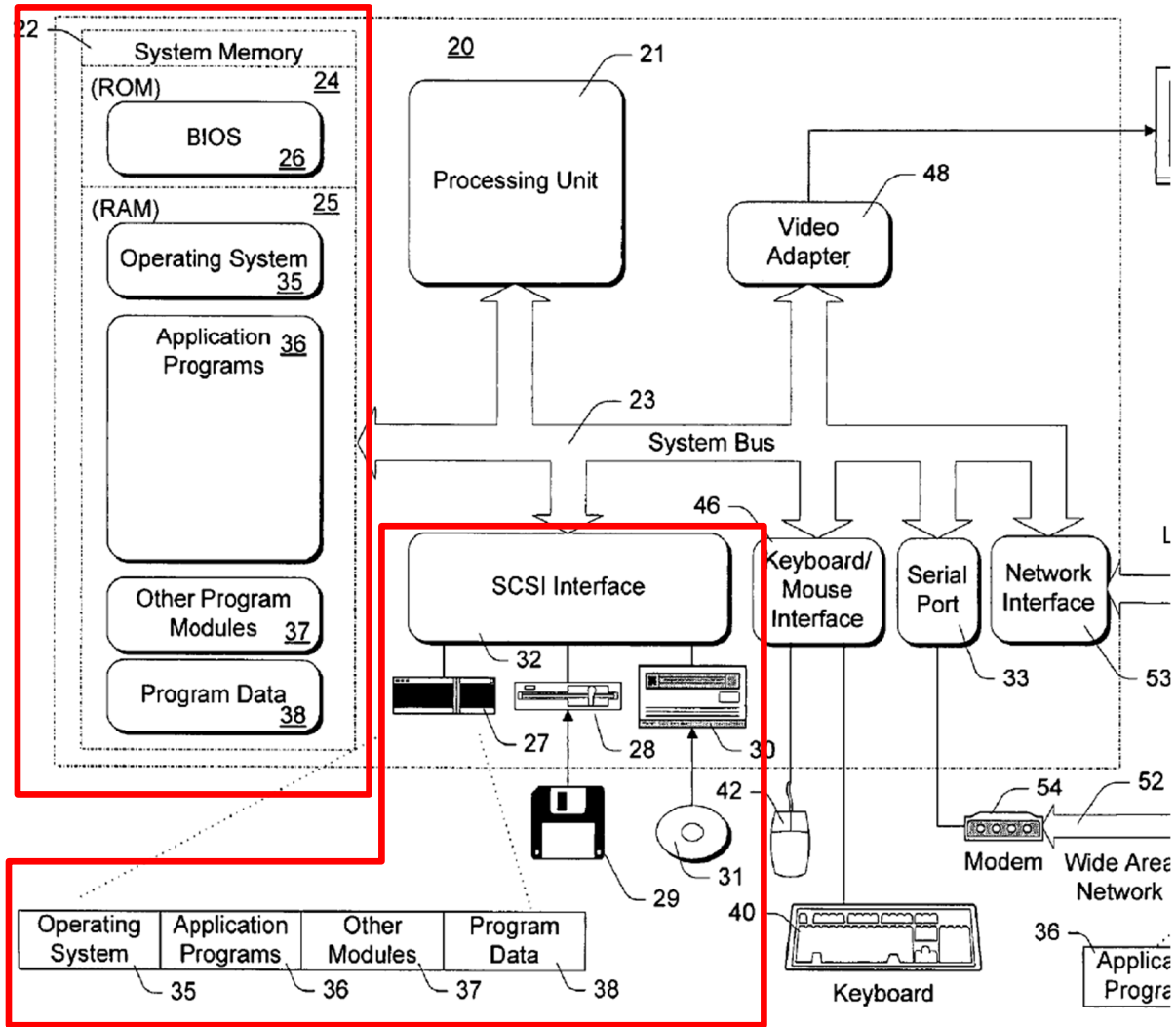
75. Gupta discloses that the computer 20 has ROM 24, RAM 25, and various “computer readable media” such as a hard disk drive 27, a magnetic disk drive 28, and an optical disk drive 30.

The bus 23 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. The system memory includes **read only memory (ROM) 24** and **random access memory (RAM) 25**. A basic input/output system (BIOS) 26, containing the basic routines that help to transfer information between elements within computer 20, such as during start-up, is

stored in ROM 24. Computer 20 further includes a **hard disk drive 27** for reading from and writing to a hard disk, not shown, a **magnetic disk drive 28** for reading from and writing to a removable magnetic disk 29, and an **optical disk drive 30** for reading from or writing to a removable optical disk 31 such as a CD ROM or other optical media. The hard disk drive 27, magnetic disk drive 28, and optical disk drive 30 are connected to the system bus 23 by an SCSI interface 32 or some other appropriate interface. The **drives** and their associated **computer-readable media** provide **nonvolatile storage of computer readable instructions, data structures, program modules and other data for computer 20**. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 29 and a removable optical disk 31, it should be appreciated by those skilled in the art that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, random access memories (RAMs) read only memories (ROM), and the like, may also be used in the exemplary operating environment.

A number of **program modules** may be **stored** on the hard disk, magnetic disk 29, optical disk 31, ROM 24, or RAM 25, including an operating system 35, **one or more application programs 36**, other **program modules 37**, and **program data 38**. ...

Id., ¶¶[0039]-[0040] (emphasis added); *see also* ¶¶[0037]-[0038], [0041]-[0042].



Id., FIG. 2 (annotated)

76. Gupta describes such at least one computer readable medium comprising a video editing environment, such as software for both editing annotations and streaming video executed by servers 10-13 and stores 17-18. For example, the annotation server 10 executes software for editing annotations.

Annotation server 10 further includes an annotation backend (ABE) module 132, which contains functionality for accessing annotation stores 17 and 18, for composing outgoing email based on annotation data, and for processing incoming email. ... Annotation server 10 is illustrated in FIG. 3 as maintaining the annotation content and associated control information (meta data) separately in two different stores. ...

Id., ¶[0048] (emphasis added).

FIG. 21 illustrates an exemplary "edit annotation" dialog box 590. Edit annotation dialog box 590 is displayed in response to a user request to edit an annotation, such as selection of the "edit" option in menu 466 of FIG. 15. Dialog box 590 includes information and fields displaying the current information for the annotation, analogous to dialog box 560 of FIG. 20. New information can be entered into the fields, analogous to add annotation dialog box 330 of FIG. 8. In the illustrated example, existing email addresses cannot be altered and new email addresses cannot be entered.

Id., ¶[0144] (emphasis added).

Dialog box 280 also includes a cancel button 318, an OK button 320, and an advanced button 322. Selection of cancel button 318 causes interface 152 to close the dialog box 280, canceling out of the add new annotation process. Selection of OK button 320 causes interface 152 to forward the information entered in dialog box 280 to annotation backend 151, which sends the

new annotation information to annotation server 10 of FIG.

3. Interface 152 then closes dialog box 280, the new annotation being entered.

Id., ¶[0100] (emphasis added).

77. Additionally, the streaming media server 11 executes software for streaming video.

Streaming **media server computer 11 has access to streaming media content** in the form of different media streams. These media streams can be individual media streams (e.g., audio, video, graphical, etc.), or alternatively composite media streams including two or more of such individual streams. Some media streams might be stored as files in a database or other file storage system, while other media streams might be supplied to the server on a "live" basis from other data source components through dedicated communications channels or through the Internet itself. Different versions of the same media content (e.g., low-resolution and high-resolution versions) may be available to server computer 11.

Id., ¶[0030] (emphasis added).

Dialog box 280 also includes a "play" button 316. Selection of play button 316 causes interface module 152 of FIG. 3 to forward a segment specification to web browser 153 of client 15. The segment specification includes the target identifier from target display 300 and the begin and end times from boxes 312 and 314,

respectively. Upon receipt of the segment specification from interface module 152, the browser communicates with media server 11 and requests the identified media segment using conventional HTTP requests. In response, **media server 11 streams the media segment to client 15 for presentation to the user.** This presentation allows, for example, the user to verify the portion of the media stream to which his or her annotation will correspond.

Id., ¶[0099] (emphasis added).

78. Gupta's video editing environment also includes the web server components and program modules at streaming media server 11 and web server 12, since the client computer accesses these servers via its browser to request relevant interfaces and data.

When a user of a client computer 15 accesses a web page containing streaming media, a conventional web browser of the **client computer 15 contacts the web server 12** to request a Hypertext Markup Language (HTML) page. The **client-based browser also submits requests to the media server 11 for streaming data**, and the annotation server 10 for any annotations associated with the streaming data. When a user of a client computer 15 desires to add or retrieve annotations, the client computer 15 contacts the annotation server 10 to perform the desired addition/retrieval.

Id., ¶[0034] (emphasis added).

79. Gupta describes such a computer system being disposed at a server side of a public network 16 (*e.g.*, Internet), in contrast to a client side of client computers

15. *Id.*, ¶[0029] (“**public network** 16 such as the **Internet**.” (emphasis added)).

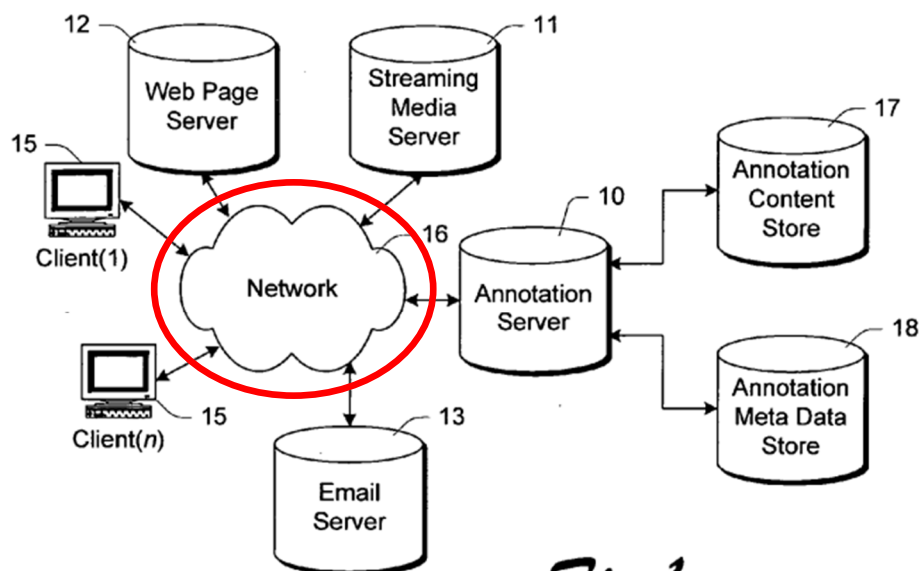


Fig. 1

Id., FIG. 1 (annotated).

3. [13.2] receiving a set of video information at the video editing environment using the computer system;

80. In my opinion, Gupta discloses limitation [13.2].

81. Gupta discloses multiple instances where the computer system at the video editing environment (*e.g.*, at the software for streaming video using servers 10-13 and stores 17-18) receives a set of video information (*e.g.*, a collection of information concerning videos).

82. A first instance of Gupta disclosing the computer system receiving a set of video information is where Gupta describes the streaming media server 11 receiving a set of videos from the video storage system or database, by virtue of having access to the video content that is stored as files in a database or other file storage system separate from the streaming media server 11.

Streaming **media server computer 11 has access to streaming media content** in the form of different media streams. These media streams can be individual media streams (e.g., audio, **video**, graphical, etc.), or alternatively composite media streams including two or more of such individual streams. Some **media streams might be stored as files in a database or other file storage system**, while other media streams might be supplied to the server on a "live" basis from other data source components through dedicated communications channels or through the Internet itself. ...

Gupta, ¶[0030] (emphasis added).

83. A second instance of Gupta disclosing the computer system receiving a set of video information is where Gupta describes the streaming media server 11 receiving “live” media streams that are “supplied” to it “from other data source components through dedicated communications channels or through the Internet itself.”

Streaming media server computer 11 has access to streaming media content in the form of different media streams. These media streams can be individual media streams (e.g., audio, video, graphical, etc.), or alternatively composite media streams including two or more of such individual streams. Some media streams might be stored as files in a database or other file storage system, while other media streams might be supplied to the server on a "live" basis from other data source components through dedicated communications channels or through the Internet itself. ...

Id., ¶[0030] (emphasis added).

84. Such video information is “received” at the video editing environment because the streaming media content is received by software executed by streaming media server 11 such as when retrieving video to stream to client 15. *Id.*, ¶[0030]. For example, FIGs. 22-23 illustrate graphical user interface windows that are running on a web browser and which display both the media stream and the corresponding annotations at the same time. For example, FIG. 23 illustrates a video playing in window 626 within a user interface of a browser web page.

FIGS. 22 and 23 illustrate exemplary graphical user interface windows that concurrently display annotations and corresponding multimedia content.

Id., ¶[0026] (emphasis added).



Fig. 22

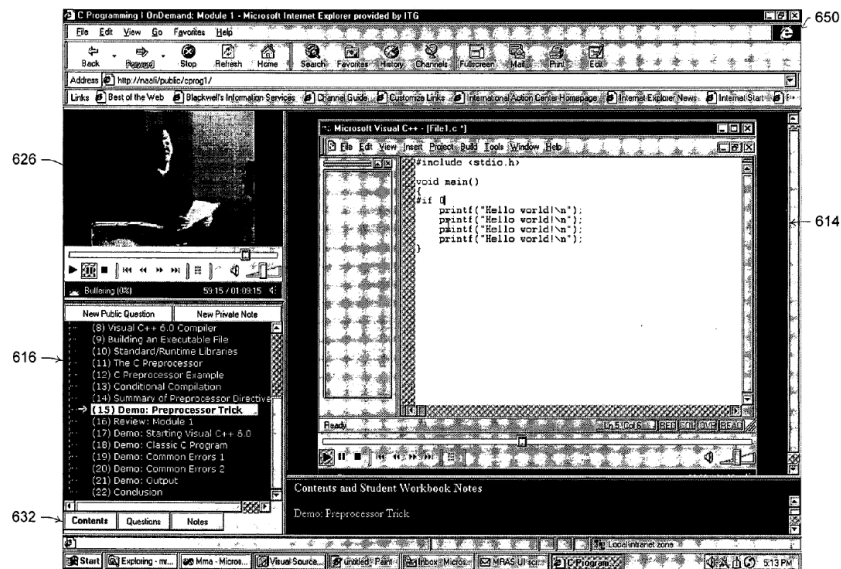


Fig. 23

Id., FIGs. 22-23.

85. Gupta also discloses that the client's access to the streaming media is provided via a client web browser requesting an HTML page from the web server 11 and submitting a request to the media server 11 to stream the video.

When a user of a client computer 15 accesses a web page containing streaming media, a conventional web browser of the client computer 15 contacts the web server 12 to request a Hypertext Markup Language (HTML) page. The client-based browser also submits requests to the media server 11 for streaming data, and the annotation server 10 for any annotations associated with the streaming data. When a user of a client computer 15 desires to add or retrieve annotations, the client computer 15 contacts the annotation server 10 to perform the desired addition/retrieval.

Id., ¶[0034] (emphasis added).

4. [13.3] receiving a set of cue point information at the video editing environment using the computer system;

86. In my opinion, Gupta discloses limitation [13.3].

87. Gupta discloses multiple instances where the computer system at the video editing environment (*e.g.*, at the software for editing annotations using servers 10-13 and stores 17-18) receives a set of cue point information (*e.g.*, a collection of information concerning annotations).

88. A first instance of Gupta's computer system (*e.g.*, annotation server 10) receiving a set of cue point information at the video editing environment is where, in providing existing annotations to a client computer, annotation server 10 receives such annotation information from an annotation meta data store 18 and annotation content store 17.

Annotation server 10 controls the storage of annotations and their provision to client computers 15. The annotation server 10 **manages the annotation meta data store 18 and the annotation content store 17.** The annotation server 10 communicates with the client computers 15 via any of a wide variety of known protocols, such as the Hypertext Transfer Protocol (HTTP). ...

Gupta, ¶[0032] (emphasis added).

Annotation server 10 further includes an annotation back end (ABE) module 132, which **contains functionality for accessing annotation stores 17 and 18,** for composing outgoing email based on annotation data, and for processing incoming email. Incoming email is received and passed to the ABE module 132 by the Email Reply Server 133. **Annotation content authored** at client 15, using user interface 152, is received by ABE 132 and **maintained in annotation content store 17.** Received meta data (control information) corresponding to the annotation content is **maintained in annotation meta data store 18.** The annotation content and meta data can be stored in any of a variety of conventional manners, such as in SQL relational databases (e.g., using Microsoft “SQL Server” version 7.0, available from Microsoft Corporation). Annotation server 10 is illustrated in FIG. 3 as maintaining the annotation content and associated control information (meta data) separately in two different stores. Alternatively, all of the annotation data (content and meta information) can be stored together in a single store, or content

may be stored by another distinct storage system on the network 16 of FIG. 1, such as a file system, media server, email server, or other data store.

Id., ¶[0048] (emphasis added).

89. Gupta's FIGs. 1 and 3 likewise illustrate the annotation server accessing annotations stored on (and therefore receiving annotation information from, for such access) an external Annotation Content Store and Annotation Meta Data Store. Gupta also discloses that, as an alternative to the Annotation Content Store and Annotation Meta Data Store, the annotation content may be stored on another storage system on the network 16 in FIG. 1.

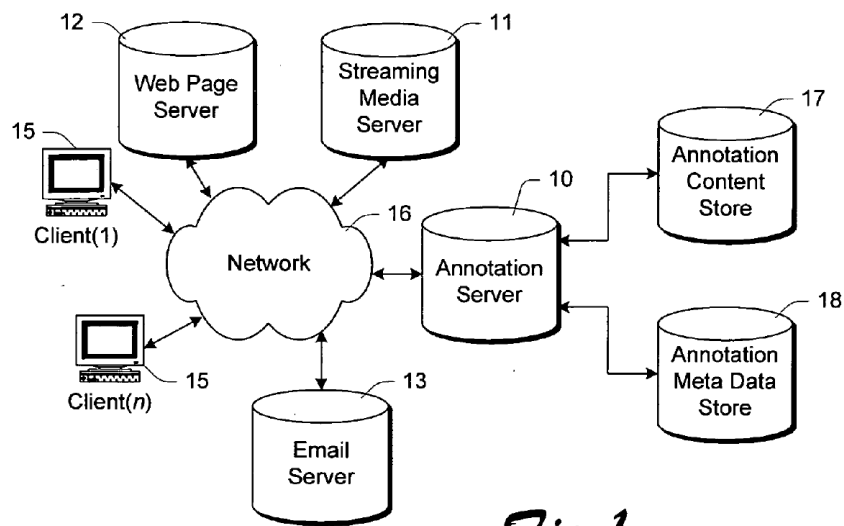


Fig. 1

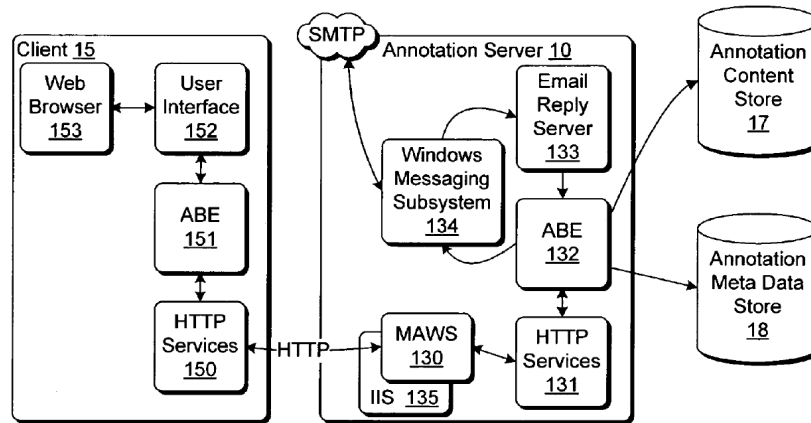


Fig. 3

Id., FIGs. 1, 3.

... Annotation server 10 is illustrated in FIG. 3 as maintaining the annotation content and associated control information (meta data) separately in two different stores. Alternatively, all of the annotation data (content and meta information) can be stored together in a single store, or content may be stored by **another distinct storage system on the network 16 of FIG. 1, such as a file system, media server**, email server, or other data store.

Id., ¶[0048] (emphasis added).

90. A second instance of Gupta's computer system (e.g., annotation server 10) receiving a set of cue point information at the video editing environment is where the annotation server 10 receives, from a client computer, information of an annotation to be added.

Annotation server 10 further includes an **annotation back end (ABE) module 132**, which contains functionality for accessing

annotation stores 17 and 18 ... **Annotation content authored at client 15, using user interface 152, is received by ABE 132 and maintained in annotation content store 17.** ...

Id., ¶[0048] (emphasis added).

91. Such cue point information (annotation information) is received at the video editing environment in Gupta because annotation information is received by the software executed by the annotation server 10, including when the annotation server 10 receives a new annotation from client 15 or retrieves an annotation for the client 15 to edit.

Annotation server 10 controls the storage of annotations and their provision to client computers 15. The annotation server 10 manages the annotation meta data store 18 and the annotation content store 17. The annotation server 10 communicates with the client computers 15 via any of a wide variety of known protocols, such as the Hypertext Transfer Protocol (HTTP). The annotation server 10 can receive and provide annotations via direct contact with a client computer 15, or alternatively via electronic mail (email) via email server 13. The annotation server 10 similarly communicates with the email server 13 via any of a wide variety of known protocols, such as the Simple Mail Transfer Protocol (SMTP).

Id., ¶[0032] (emphasis added).

Annotation server 10 further includes an annotation backend (ABE) module 132, which contains functionality for accessing annotation stores 17 and 18, for composing outgoing email based on annotation data, and for processing incoming email. Incoming email is received and passed to the ABE module 132 by the Email Reply Server 133. Annotation content authored at client 15, using user interface 152, is received by ABE 132 and maintained in annotation content store 17. Received meta data (control information) corresponding to the annotation content is maintained in annotation meta data store 18. ...

Id., ¶[0048] (emphasis added).

Dialog box 280 also includes a cancel button 318, an OK button 320, and an advanced button 322. Selection of cancel button 318 causes interface 152 to close the dialog box 280, canceling out of the add new annotation process. Selection of OK button 320 causes interface 152 to forward the information entered in dialog box 280 to annotation backend 151, which sends the new annotation information to annotation server 10 of FIG. 3. Interface 152 then closes dialog box 280, the new annotation being entered.

Id., ¶[0100] (emphasis added).

5. [13.4] receiving a cue point editing command over the network from a remote client terminal disposed at a client side of the public network using the computer system;

92. In my opinion, Gupta discloses limitation [13.4].

93. Gupta discloses the computer system receiving a cue point editing command (*e.g.*, command to edit an annotation after a client selects an “OK” dialog button in the FIG. 21 interface) from a remote client terminal (*e.g.*, client 15). For example, Gupta describes that to edit an existing annotation, a client user enters an edit mode by selecting an existing annotation (*e.g.*, displayed in annotation listing box 446) to bring up a selection box/menu 466 having an “edit” option which, when selected, causes an interface module 152 to display another dialog box allowing the user to edit the content of the annotation.

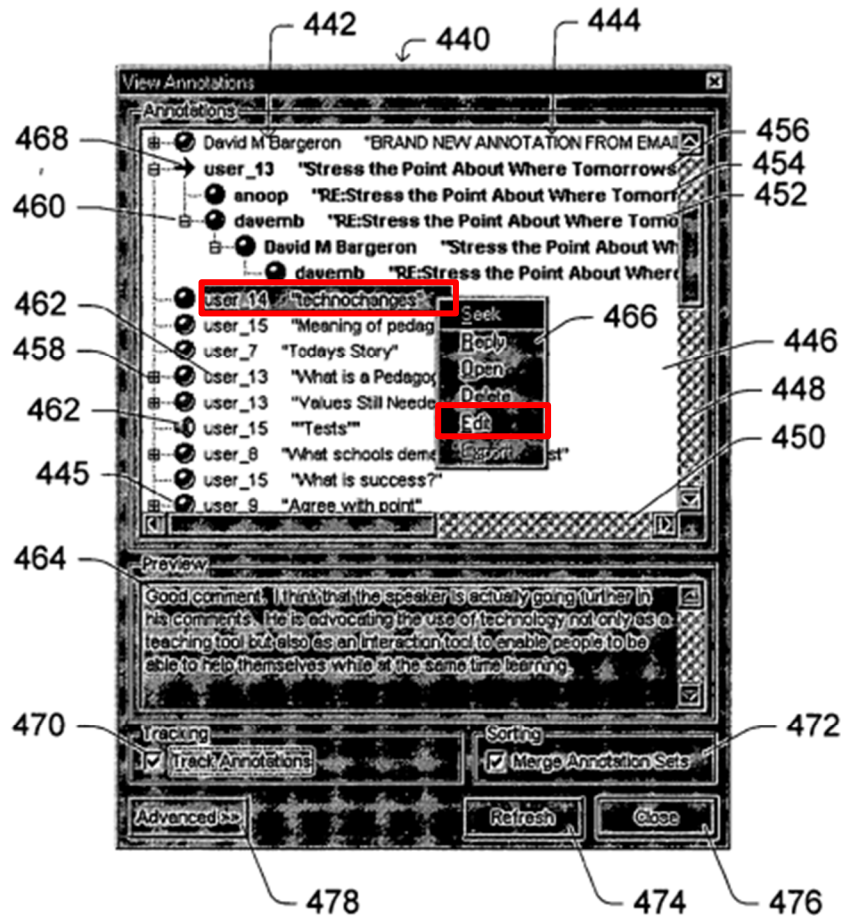


Fig. 15

Id., FIG. 15.

FIG. 15 shows a dialog box 440 that identifies annotations corresponding to one or more annotation sets. The identified annotations are the result of the query input by the user as discussed above with reference to FIGS. 12-14, or alternatively by selection of show annotations button 266 of FIG. 6. In the illustration of FIG. 15, annotation identifiers in form of user identifiers 442 and subject lines 444 are displayed within an

annotation listing box 446. The user can scroll through annotation identifiers in a conventional manner via scroll bars 448 and 450. The annotation identifiers are presented in annotation listing box 446 according to a default criteria, such as chronological by creation time/date, by user, alphabetical by summaries, etc. Icons identifying the type of annotation are also displayed in listing box 446, such as a circle 445 to represent text and a microphone 447 to represent audio.

...

By selecting a particular annotation identifier from annotation listing 446 (e.g., “single clicking” on the summary using the left mouse button), preview information is presented in a preview section 464, and **a selection box or menu 466 is provided.** The exact nature of the preview information is dependent on the amount of information that was requested (e.g., as identified in level of detail 422 of FIG. 14).

Selection box 466 provides various user-selectable features related to the annotation whose identifier is selected, including “seek”, “reply”, open “delete”, and **“edit”**. Selection of the “seek” feature causes interface module 152 of FIG. 3 to initiate presentation of the media segment corresponding to the annotation (analogous to the play button 316 of FIG. 7 discussed above). ...

... **Selection of the “edit” feature** causes interface module 152 to **display another dialog box allowing the user to edit the content of the annotation,** as discussed in more detail below. ...

Id., ¶¶[0124]-[0128] (emphasis added).

94. Gupta discloses that after a user is provided with the annotation editor interface (*e.g.*, “‘edit annotation’ dialog box 590”), new information can be entered into the fields within the dialog box 590, in an analogous manner as the annotation dialog box 330 in FIG. 8. After editing the annotation information (*e.g.*, “set,” “subject,” and “text”, along with additional options available by expanding the “Advanced” button), the user selects “OK,” and this user selection results in a command to edit the annotation being sent from client 15 and received using the computer system’s annotation server 10.

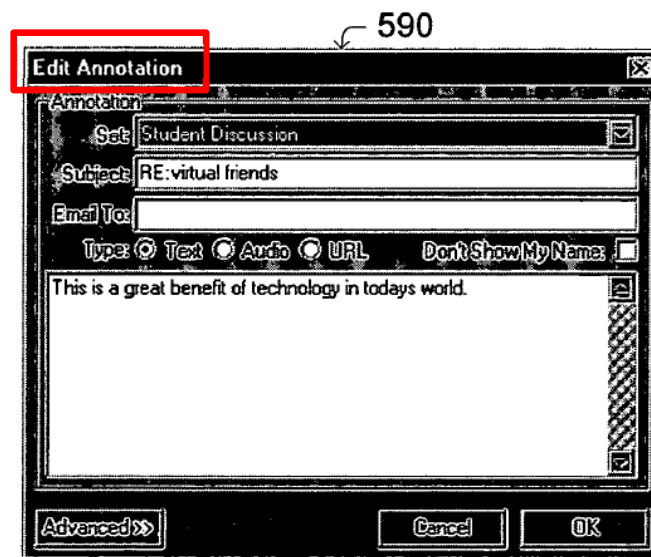


Fig. 21

Id., FIG. 21 (annotated).

FIG. 21 illustrates an exemplary “edit annotation” dialog box 590. Edit annotation dialog box 590 is displayed in response to a user request to edit an annotation, such as selection of the “edit” option in menu 466 of FIG. 15. Dialog box 590 includes information and fields displaying the current information for the annotation, analogous to dialog box 560 of FIG. 20. **New information can be entered into the fields, analogous to add annotation dialog box 330 of FIG. 8. ...**

Id., ¶[0144] (emphasis added).

Dialog box 560 also includes an advanced button 572 and close button 574. **Advanced button 572** provides **additional information regarding the annotation**, such as an identifier of the corresponding media content, a display of the temporal range of the media content that the annotation corresponds to, etc. Close button 574 causes interface module 152 to terminate display of dialog box 560.

Id., ¶[0143] (emphasis added).

Dialog box 280 also includes a cancel button 318, an OK button 320, and an advanced button 322. Selection of cancel button 318 causes interface 152 to close the dialog box 280, canceling out of the add new annotation process. **Selection of OK button 320** causes interface 152 to forward the information entered in dialog box 280 to annotation backend 151, which **sends the new annotation information to annotation server 10 of FIG. 3.**

Interface 152 then closes dialog box 280, the new annotation being entered.

Id., ¶[0100] (emphasis added).

95. Based at least on Gupta's reference to editing annotations being "analogous" to adding annotations as shown in FIG. 8 as noted above, a POSITA would therefore have understood and recognized Gupta's disclosures involving its "add annotation" functionality to extend in the same way to its disclosures regarding its "edit annotation" functionality. Such extension in common functionality includes the operations that result from when the "OK" button is selected, including the resulting forwarding of new/updated annotation information to the annotation server 10 (as Gupta describes in ¶[0100]) and the client's contacting of the annotation server 10 to perform a desired editing/adding of an annotation (as Gupta describes in ¶[0034]).

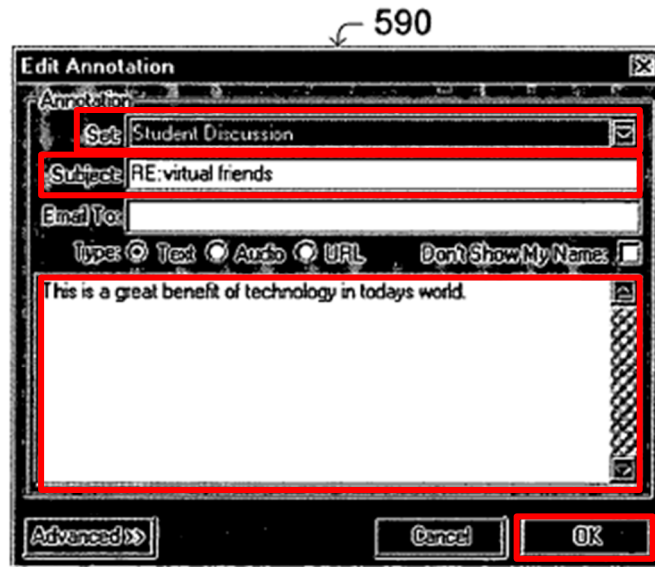


Fig. 21

Id., Fig. 21 (annotated, showing “set,” “subject,” “text” editable fields, an “advanced” button to access additional editable fields, and an “OK” button for saving changes).

Dialog box 280 also includes a cancel button 318, an OK button 320, and an advanced button 322. Selection of cancel button 318 causes interface 152 to close the dialog box 280, canceling out of the add new annotation process. **Selection of OK button 320** causes interface 152 to forward the information entered in dialog box 280 to annotation backend 151, which **sends the new annotation information to annotation server 10** of FIG. 3. Interface 152 then closes dialog box 280, the new annotation being entered.

Id., ¶[0100] (emphasis added).

When a user of a client computer 15 accesses a web page containing streaming media, a conventional web browser of the client computer 15 contacts the web server 12 to request a Hypertext Markup Language (HTML) page. The client-based browser also submits requests to the media server 11 for streaming data, and the annotation server 10 for any annotations associated with the streaming data. **When a user of a client computer 15 desires to add or retrieve annotations**, the client computer 15 **contacts the annotation server 10 to perform the desired addition/retrieval**.

Id., ¶[0034] (emphasis added).

96. Therefore, the user's selection of "OK" after changing annotation information (*e.g.*, "set," "subject," "text") in the FIG. 21 interface results in a cue point editing command that is sent from the client computer 15 to the annotation server 10, and is therefore received at the computer system.

97. Gupta describes the computer system's receipt of such command over the network (*e.g.*, Internet 16). The client 15 is remote from the servers 10-12 because such intercommunication is over the Internet.

Client 15 also executes a web browser module 153, which provides a conventional web browsing interface and capabilities for the user **to access various servers via network 16 of FIG. 1**. Web browser 153 also provides the interface for a user to select particular media streams for presentation. The user can select

which one of different versions of multimedia content he or she wishes to receive from media server 11 of FIG. 1. This selection can be direct (e.g., entry of a particular URL or selection of a “low resolution” option), or indirect (e.g., entry of a particular desired playback duration or an indication of system capabilities, such as “slow system” or “fast system”). Alternatively, other media presentation interfaces could be used.

Id., ¶[0046] (emphasis added); *see also* FIG. 1, ¶[0029] (network 16 as Internet).

98. A POSITA would have understood the plain meaning of “cue point editing command” to be a command, sent from the remote client terminal and received by the computer system, to edit a portion of the cue point information. *See* limitation [13.5] (indicating that “according to the cue point editing command” “a portion of the cue point information” is “modif[ied]”) (emphasis added). Further, a POSITA would have understood the word “command” (in “cue point editing command”) to mean an instruction to a computer program, due to how this word is used in the ’286 Patent.

99. The ’286 Patent describes that “the command, ‘RETRIEVE (‘VID—0001’), is used to retrieve a video file identified as ‘VID—0001’ from the video server 112. When the video server 112 receives the ‘RETRIEVE’ command, a relational database may be queried to find an entry for ‘VID—0001’ ... The video server 112 may then use various algorithms to select a download location and

download the file.” ’286 Patent, 7:33-41 (emphasis added). In other words, an instruction is sent to the video server to retrieve a video, to which the video server responds. *See also id.*, 13:65-67 (describing “Visual Basic **commands** for an interpreter” that are instructions to the interpreter “to generate the GUI”) (emphasis added), 42:45-48 (describing a “**command** 1640” “transmitted by a client, received by a host or application” “requesting that a dataset is built for all ‘Exterior’ cue points” and the host or application responds to the instruction by “generat[ing] dataset 1650.”) (emphasis added).

100. This understanding, that a “command” is an instruction to a computer program, also comports with contemporaneous technical dictionaries, *e.g.*, the Microsoft Computer Dictionary defines a “command” as “An instruction to a computer program that, when issued by the user, causes an action to be carried out.” Exhibit 1017.

101. Therefore, a POSITA would have understood that the user’s selection of “OK” after changing annotation information (*e.g.*, “set,” “subject,” “text”) in the FIG. 21 interface results in sending a cue point editing command—an instruction to the computer system to edit a portion of the annotation—because the computer system responds by carrying out the requested action (*see* limitation [13.5] below).

102. Gupta also describes client 15 being disposed at a client side of the public network 16, as distinguished from the server side containing servers 10-12.

Notably, Gupta discloses the system as a “client/server network system and environment,” which therefore is arranged according to a client/server architecture.

FIG. 1 shows a **client/server network system and environment** in accordance with the invention. Generally, the system includes multiple network server computers 10, 11, 12, and 13, and multiple (n) network client computers 15. The computers communicate with each other over a data communications network. The communications network in FIG. 1 comprises a public network 16 such as the Internet. The data communications network might also include, either in addition to or in place of the Internet, local-area networks and/or private wide-area networks.

Id., ¶[0029] (emphasis added), FIG. 1.

6. [13.5] modifying a portion of the cue point information at the server side of the public network according to the cue point editing command received over the public network from the client terminal;

103. In my opinion, Gupta discloses limitation [13.5].

104. As an initial matter, I note that the claim language merely recites “modifying a portion of the cue point information” and does not require saving the modified information or replacing the prior cue point information with the newly modified information. *See* ’286 Patent, Claim 13; 19:26-35 (distinguishing “modifying” from “adding” and “deleting”). Therefore, a POSITA would have

understood limitation [13.4] as requiring only modifying existing cue points in accordance with the received cue point editing command and nothing further.

105. Gupta discloses the server side of the public network (*e.g.*, annotation server 10) modifying the at least a portion of the cue point information (annotation information) according to the cue point editing command received during limitation [13.4]. In Gupta, when the user of a client computer 15 makes modifications to annotation information via the “Edit Annotation” dialog box and selects “OK,” the dialog box information is sent over the network to modify the edited annotation at the annotation server 10. This is because Gupta describes the operation of the “Edit Annotation” dialog box of FIG. 21 to work in the same manner as the “Add Annotation” dialog boxes illustrated in FIGs. 7 and 8 in which annotation information is modified based on the information received from the client.

FIG. 21 illustrates an exemplary “edit annotation” dialog box 590. Edit annotation dialog box 590 is displayed in response to a user request to edit an annotation, such as selection of the “edit” option in menu 466 of FIG. 15. Dialog box 590 includes information and fields displaying the current information for the annotation, analogous to dialog box 560 of FIG. 20. **New information can be entered into the fields, analogous to add annotation dialog box 330 of FIG. 8. ...**

Gupta, ¶[0144] (emphasis added).

Dialog box 280 [of FIG. 7] also includes a cancel button 318, an OK button 320, and an advanced button 322. ... **Selection of OK button 320** causes interface 152 to forward the information entered in dialog box 280 to annotation backend 151, which **sends the new annotation information to annotation server 10 of FIG. 3**....

FIG. 8 shows another exemplary “add new annotation” dialog box 330. **Dialog box 330 is similar to dialog box 280 of FIG. 7**, but is a reduced option box. As illustrated, **dialog box 330 includes much of the same information and fields as dialog box 280 of FIG. 7**.

Id., ¶¶[0100]-[0102] (emphasis added).

Annotation server 10 further includes an annotation back end (ABE) module 132, which contains functionality for accessing annotation stores 17 and 18 ... Annotation content **authored at client 15**, using user interface 152, is **received by ABE 132** and maintained in annotation content store 17. ...

Id., ¶[0048] (emphasis added).

... When a user of a client computer 15 **desires to add or retrieve annotations**, the client computer 15 **contacts the annotation server 10** to **perform the desired addition/retrieval**.

Id., ¶[0034] (emphasis added).

106. Further, Gupta discloses the modifications taking place at the annotation server 10, which is at the server side of the public network (Internet). *Id.*, ¶¶[0029] (server side), [0034] (“annotation server 10 ... perform[s] the desired addition/retrieval”).

7. [13.6] receiving a cue point selection over the network from the remote client terminal identifying one of the cue points corresponding to a portion of the video information of one of the video files; and

107. In my opinion, Gupta discloses limitation [13.6].

108. Gupta discloses receiving a cue point selection (*e.g.*, receiving a user input of a selected annotation via a request to “seek” or “navigate” to that annotation) from the remote client terminal (*e.g.*, client 15), including when a user selects “seek” in the menu 466 for a selected annotation in user interface 152 or selects “navigate” in the dialog box 520 of the FIG. 17 interface.

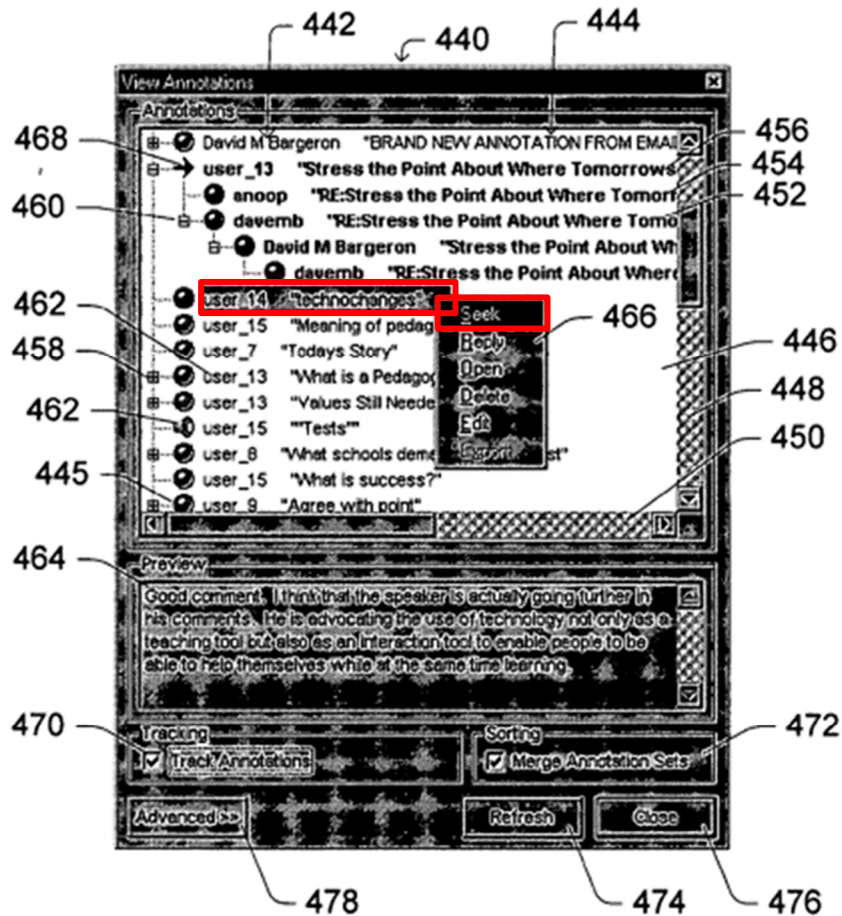


Fig. 15

Gupta, FIG. 15; *see also* FIGs. 16, 22-23.

Target preferences 494 include seek options 498 and checkbox 500. Seek options 498 allow the user to select from various options the resultant behavior from seeking to an annotation (e.g., by selecting “seek” in menu 466 of FIG. 15). Additional options can be selected via a drop-down or pull-down menu by actuating icon 502. Examples of options include: “seek and play to end”, which initiates playback of the multimedia content at

the temporal position corresponding to the begin time of the annotation and continuing until the end of the multimedia content; “seek and play annotation”, which plays back only the temporal segment of the multimedia content identified by the begin and end times of the annotation; “seek and skip to next”, which plays the temporal segment of the multimedia content identified by the begin and end times of the annotation, then jumps to play the temporal segment of the next annotation in annotation listing 446 (and continually jumping to the next temporal segment when the current temporal segment end time is reached). Dialog box 490 also includes an advanced button 504, which causes interface module 152 to display a view annotations dialog box having less information and/or options, such as dialog box 440 of FIG. 15.

Id., ¶[0135] (emphasis added); *see also* ¶[0127], [0136] (describing the “navigate” option).

A user can browse through the visual content of media portion 612 using shuttle controls 628 or by moving “thumb” 638 along slider bar 640, which automatically synchronizes the display of the slides in annotation content 614 and the identifiers in annotation identifier portion 616 with the current seek point in the video. Alternatively, double-clicking on or “seeking” to any of the annotations (via the identifiers in annotation identifier portion 616) can be used as a navigational tool as well.

Id., ¶[0152] (emphasis added).

109. Gupta describes that the user's selection of the "seek" option "causes interface module 152 of FIG. 3 to initiate presentation of the media segment corresponding to the annotation, and such operation is "analogous" to the user operating the play button 316.

Selection box 466 provides various user-selectable features related to the annotation whose identifier is selected, including "seek", "reply", open "delete", and "edit". Selection of the "seek" feature causes interface module 152 of FIG. 3 to initiate presentation of the media segment corresponding to the annotation (analogous to the play button 316 of FIG. 7 discussed above). ...

Id., ¶[0127] (emphasis added).

110. Additionally, Gupta describes that the play button 316 involves the client browser sending a HTTP request identifying the user-specified annotation (and the corresponding media segment) to the media server 11 to play that media segment—therefore, the media server 11 receives a cue point section with the identified annotation.

Dialog box 280 also includes a "play" button 316. Selection of play button 316 causes interface module 152 of FIG. 3 to forward a segment specification to web browser 153 of client 15. The segment specification includes the target identifier from target display 300 and the begin and end times from boxes 312 and 314, respectively. Upon receipt of the segment specification from

interface module 152, the browser communicates with media server 11 and requests the identified media segment using conventional HTTP requests. In response, media server 11 streams the media segment to client 15 for presentation to the user. This presentation allows, for example, the user to verify the portion of the media stream to which his or her annotation will correspond.

Id., ¶[0099] (emphasis added).

Therefore, the media server 11 receives the cue point selection.

111. Additionally, such receipt of the cue point selection is over the network (e.g., Internet), as the client “access[es] various servers via network 16,” including media server 11. *Id.*, ¶[0046], Fig. 1.

112. Finally, the received cue point selection identifies one of the cue points corresponding to a portion of the video information in one of the video files. As I explain above with respect to limitation [13.3], each annotation (cue point) corresponds to a portion (temporal portion with beginning and end) of the video information of a video file in the system. *See* my discussion above of limitation [13.3]; *see also*:

Multimedia presentations may also include “annotations” relating to the multimedia presentation. An annotation is data (e.g., audio, text, video, etc.) that corresponds to a multimedia presentation. ... These annotations typically correspond to a

particular temporal location in the multimedia presentation

and can provide a replacement for much of the “in-person” interaction and “classroom discussion” that is lost when the presentation is not made “in-person” or “live”.

Gupta, ¶[0007] (emphasis added).

Time range field 184 contains data representing **“begin” and “end” times defining a segment of media timeline to which annotation entry 180 is associated.** Time units field 186 contains data representing the units of time represented in time range field 184. Together, **time range field 184 and time units field 186 identify the relative time range of the annotation represented by annotation entry 180.** This relative time range **corresponds to a particular segment of the media content to which annotation entry 180 is associated.** The begin and end times for the annotation are provided by the user via interface 152 of FIG. 3, or alternatively can be automatically or implicitly derived using a variety of audio and video signal processing techniques, such as sentence detection in audio streams or video object tracking.

Id., ¶[0053] (emphasis added).

Streaming media server computer 11 has access to streaming media content in the form of different media streams. ... Some media streams might be **stored as files** in a database or other file storage system, ...

Id., ¶[0030] (emphasis added).

113. Therefore, a user's selection of "seek" for a particular annotation identifies the desired cue point. Gupta, ¶¶[0007], [0053], [0135].

8. [13.7] streaming the portion of the video information to the client terminal over the public network in response to receiving the cue point selection.

114. In my opinion, Gupta discloses limitation [13.7].

115. Gupta discloses, in response to receiving the cue point (annotation) selection ("seek" or "navigate") (*see* my discussion above of limitation [13.6]), streaming the associated portion of the video information to the client terminal (*e.g.*, client 15) over the public network 16 (*e.g.*, Internet). As I explain above, a user selecting the "seek" option for an annotation "initiate[s] presentation of the media segment corresponding to the annotation" and is "analogous to the play button 316" (Gupta, ¶[0127]), which involves "the [client] browser communicat[ing] with media server 11 and request[ing] the identified media segment using conventional HTTP requests" and in response, the media server 11 streams the media segment to client 15 for presentation to the user. The "navigate" option has similar functionality. *See* my discussion above of limitation [13.6]); *see also*:

Dialog box 280 also includes a "play" button 316. Selection of play button 316 causes interface module 152 of FIG. 3 to forward a segment specification to web browser 153 of client 15. The segment specification includes the target identifier from target display 300 and the begin and end times from boxes 312 and 314,

respectively. Upon receipt of the segment specification from interface module 152, the browser communicates with media server 11 and requests the identified media segment using conventional HTTP requests. **In response, media server 11 streams the media segment to client 15 for presentation to the user.** This presentation allows, for example, the user to verify the portion of the media stream to which his or her annotation will correspond.

Id., ¶[0099] (emphasis added).

The term "streaming" is used to indicate that the data representing the various media types is provided over a network to a client computer on a real-time, as-needed basis, rather than being pre-delivered in its entirety before playback. Thus, the client computer renders streaming data as it is received from a network server, rather than waiting for an entire "file" to be delivered.

Id., ¶[0006].

Streaming media server computer 11 has access to **streaming media content** in the form of different media streams. These media streams can be individual media streams (e.g., audio, video, graphical, etc.), or alternatively composite media streams including two or more of such individual streams. Some media streams might be stored as files in a database or other file storage system, while other media streams **might be supplied to the server on a "live" basis** from other data source components

through dedicated communications channels or through the Internet itself. Different versions of the same media content (e.g., low-resolution and high-resolution versions) may be available to server computer 11.

Id., ¶[0030] (emphasis added).

The annotations managed by annotation server 10 correspond to the **streaming media available from media server computer 11**. In the discussions to follow, the annotations are discussed as corresponding to streaming media. However, it should be noted that the annotations can similarly correspond to "pre-delivered" rather than streaming media, such as media previously stored at the client computers 15 via the network 16, via removable magnetic or optical disks, etc.

When a user of a **client computer 15 accesses a web page containing streaming media**, a conventional web browser of the client computer 15 contacts the web server 12 to request a Hypertext Markup Language (HTML) page. The client-based browser also **submits requests to the media server 11 for streaming data**, and the annotation server 10 for any annotations associated with the streaming data. When a user of a client computer 15 desires to add or retrieve annotations, the client computer 15 contacts the annotation server 10 to perform the desired addition/retrieval.

Id., ¶¶[0033]-[0034] (emphasis added).

... Additional options can be selected via a drop-down or pull-down menu by actuating icon 502. Examples of options include: "seek and play to end", which **initiates playback of the multimedia content at the temporal position** corresponding to the begin time of the annotation and continuing until the end of the multimedia content; "seek and play annotation", which **plays back only the temporal segment of the multimedia content identified by the begin and end times of the annotation**; "seek and skip to next", which plays the temporal segment of the multimedia content identified by the begin and end times of the annotation, then jumps to play the temporal segment of the next annotation in annotation listing 446 (and continually jumping to the next temporal segment when the current temporal segment end time is reached). ...

[0136] FIG. 17 shows another exemplary "view annotations" dialog box 520. Dialog box 520 includes much of the same information for viewing annotations as dialog box 440 of FIG. 15. However, interface module 152 has disabled the merge annotation sets 472 (which causes interface module 152 to re-render the annotations so that the annotations from the same set are displayed grouped together under a tree item with the name of the set as its title, such as "Student Discussion") and track annotations 470 options. Additionally, menu 522 includes different options than those included in menu 466 of FIG. 15. Specifically, menu 522 includes a **"navigate"** option rather than a "seek" option. **The "navigate" option is similar to the "seek"**

option discussed above in menu 466, however the "navigate" option is used when the annotation does not correspond to the current media content being targeted by client 15 (FIG. 3). Selection of the "navigate" option causes user interface module 152 to instruct web browser 153 to navigate to the media content to which the annotation corresponds, making that the current targeted media content.

Id., ¶¶[0135]-[0136] (emphasis added).

C. Claim 14

116. Claim 14 recites:

14. The method of claim 13, further comprising providing a cue point editing mode as part of the video editing environment configured to perform at least one of creating, modifying, deleting, or interpreting at least a portion of the set of cue point information according to the cue point editing command received over the public network from the client terminal.

117. In my opinion, Claim 14 is anticipated by Gupta.

118. Gupta discloses providing a cue point (annotation) editing mode as part of the video editing environment, which includes providing interfaces for editing annotations as part of the software executed by servers 10-13 and stores 17-18, such as to perform operations at the annotation server 10 when a user selects “edit” for a selected annotation in the FIG. 15 interface to display the “Edit Annotation” dialog box. *See my discussion above of limitation [13.4]; see also:*

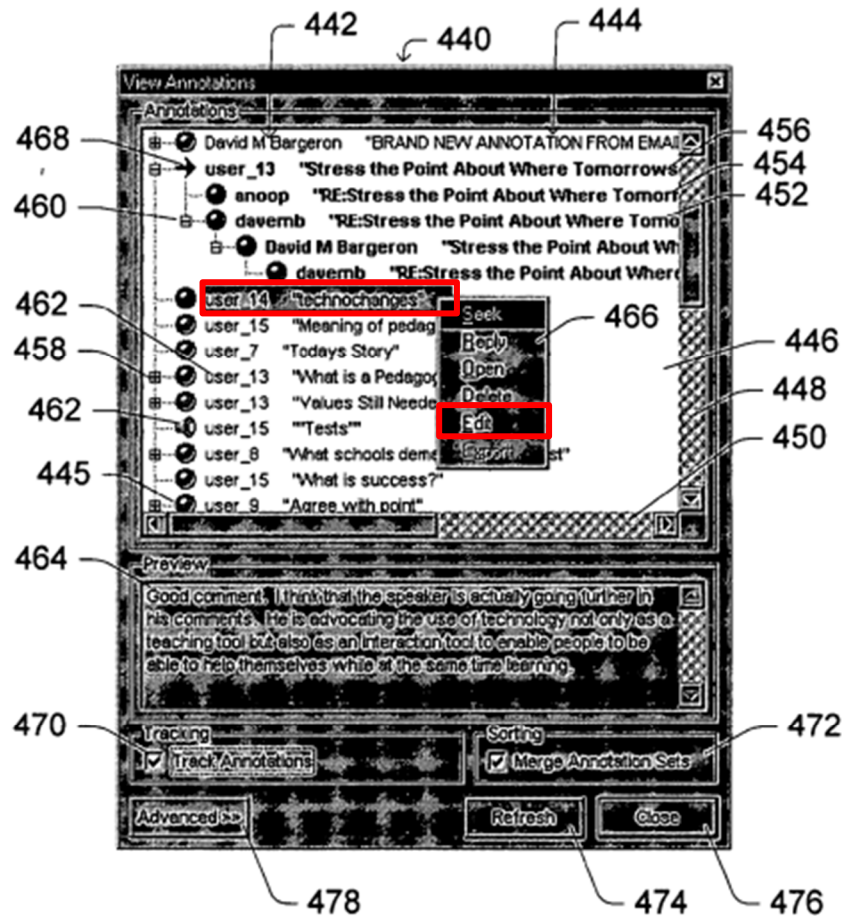


Fig. 15

Id., FIG. 15.

FIG. 15 shows a dialog box 440 that identifies annotations corresponding to one or more annotation sets. The identified annotations are the result of the query input by the user as discussed above with reference to FIGS. 12-14, or alternatively by selection of show annotations button 266 of FIG. 6. In the illustration of FIG. 15, annotation identifiers in form of user identifiers 442 and subject lines 444 are displayed within an

annotation listing box 446. The user can scroll through annotation identifiers in a conventional manner via scroll bars 448 and 450. The annotation identifiers are presented in annotation listing box 446 according to a default criteria, such as chronological by creation time/date, by user, alphabetical by summaries, etc. Icons identifying the type of annotation are also displayed in listing box 446, such as a circle 445 to represent text and a microphone 447 to represent audio.

...

By selecting a particular annotation identifier from annotation listing 446 (e.g., “single clicking” on the summary using the left mouse button), preview information is presented in a preview section 464, and **a selection box or menu 466 is provided.** The exact nature of the preview information is dependent on the amount of information that was requested (e.g., as identified in level of detail 422 of FIG. 14).

Selection box 466 **provides various user-selectable features related to the annotation whose identifier is selected,** including “seek”, “reply”, open “delete”, and **“edit”**. Selection of the “seek” feature causes interface module 152 of FIG. 3 to initiate presentation of the media segment corresponding to the annotation (analogous to the play button 316 of FIG. 7 discussed above). ...

... Selection of the **“edit”** feature causes interface module 152 to **display another dialog box allowing the user to edit the content of the annotation,** as discussed in more detail below. ...

Id., ¶¶[0124]-[0128] (emphasis added).

119. As I explain above, Gupta discloses its annotation server 10 modifying at least a portion of the annotation information according to the cue point editing command received over the public network (Internet) from the client terminal (client 15). *See* my discussion above of limitations [13.4]-[13.5], explaining how Gupta’s “Edit Annotation” dialog box has editable fields for selected annotation including: “set,” “subject,” “text”; and an “OK” button (cue point editing command) for saving changes made in “Edit Annotation” dialog box and sending the editing command to modify the annotation information at annotation server 10, which in turn modifies the annotation.

D. Claim 15

120. Claim 15 recites:

15. The method of claim 14, further comprising: receiving an edit mode entry signal over the network from the client terminal representing entry into the cue point edit mode, and switching the video editing environment to the cue point editing mode in response to receiving the edit mode entry signal.

121. In my opinion, Claim 15 is anticipated by Gupta.

122. Gupta discloses receiving an edit mode entry signal (*e.g.*, signal for entry to the interface(s) to edit an annotation after selecting “edit” from the menu 466) over the network from the client terminal (client 15) representing entry into the

cue point edit mode (e.g., representing entry into the interface(s) for editing annotations), and switching the video editing environment to the cue point editing mode by switching the software executed by servers 10-13 and stores 17-18 to display the “Edit Annotation” dialog box in response to receiving the edit mode entry signal.

123. As I explain above, Gupta describes annotation server 10 having “web server functionality” including a “HTTP Services module 131 which manages communication with client 15.” *See* my discussion above of limitation [13.4] and claim 14; *see also*:

Selection box 466 provides various **user-selectable features** related to the **annotation whose identifier is selected, including** "seek", "reply", open "delete", and **"edit"**. ...

... **Selection of the "edit" feature** causes interface module 152 to **display another dialog box** allowing the **user to edit the content of the annotation**, as discussed in more detail below. ...

Gupta, ¶¶[0127]-[0128] (emphasis added); *see also* ¶¶[0047], [0082] (Gupta teaching that the client’s annotation user interface “may be incorporated into the web browser”).

124. Web server interaction involves a client sending an HTTP request to the web server, and the web server responding to the request by sending HTML information to render a web page interface on the client’s web browser. *Id.*, ¶¶[0044] (“commands are formulated at client computer 15 and forwarded to annotation

server 10 **via HTTP requests**” (emphasis added)), [0034] (generally describing web page request operations). *See also* my discussion above of limitation [13.4] and claim 14.

125. Therefore, in Gupta, when a user selects the “edit” option for a selected annotation, annotation server 10 receives an edit mode entry signal to enter a cue point edit mode and provide HTML pages to display the “Edit Annotation” dialog box on the client computer 15’s web browser.

126. A POSITA would have understood the plain meaning of “edit mode entry signal” to be a signal, sent from the remote client terminal and received by the computer system, to enter an edit mode. Further, a POSITA would have understood the word “signal” (in “edit mode entry signal”) to mean an indication. *See* Exhibit 1018 (“Signal.” Merriam-Webster.com Dictionary, Merriam-Webster, <https://www.merriam-webster.com/dictionary/signal> (accessed March 29, 2024) (defining “signal” as “1: SIGN, INDICATION”)).

127. As the ’286 Patent explains, “in some embodiments, cue point edit mode will always be available, while other embodiments may require **active entry into that mode, for example by polling for input.**” ’286 Patent, 15:28-31 (emphasis added). The “input” may be “a particular event from an input device, like a mouse or keyboard.” *Id.*, 15:31-33; *see also id.*, 18:42:49 (explaining “the client may enter the cue point edit mode in any useful way” including “click[ing] on a

video image” or “select[ing] a GUI control dedicated to providing entry into the cue point edit mode” or “select[ing] a portion of the video progress bar”).

128. Therefore, a POSITA would have understood that when a user selects the “edit” option in menu 466 for a selected annotation, annotation server 10 receives an edit mode entry signal—an indication to the computer system to enter an edit mode.

E. Claim 18

129. Claim 18 recites:

18. The method of claim 13, further comprising receiving, over the network, authentication information relating to access authority of the remote client terminal.

130. In my opinion, Claim 18 is anticipated by Gupta.

131. Gupta discloses receiving authentication information relating to access authority of the remote client terminal (*e.g.*, client 15), in the form of read and write access controls establishing which users are able to read and/or write to the annotation set.

Dialog box 280 includes an annotation set identifier 282, a subject line 284, and an email field 286. Annotation set identifier 282 allows the user to identify a named set to which the new annotation will belong. This set can be a previously defined set, or a new set being created by the user. Selection of the particular set can be made from a drop-down menu activated by selection

of icon 288, or alternatively can be directly input by the user (e.g., typed in using an alphanumeric keyboard). According to one embodiment of the invention, **annotation server 10 of FIG. 3 supports read and write access controls, allowing the creator of the set to identify which users are able to read and/or write to the annotation set.** In this embodiment, **only those sets for which the user has write access can be entered** as set identifier 282. Alternatively, rather than (or in addition to) user selection of an annotation set, one or more annotation sets may be inherently associated with dialog box 280 (or toolbar 250 of FIG. 6) and thus implicitly selected for each new annotation.

Gupta, ¶[0090] (emphasis added).

Multimedia presentations may also include “annotations” relating to the multimedia presentation. An annotation is data (e.g., audio, text, video, etc.) that corresponds to a multimedia presentation. **Annotations can be added** by anyone **with appropriate access rights** to the annotation system (e.g., the lecturer/trainer or any of the students/trainees). ...

Id., ¶[0007] (emphasis added).

Dialog box 370 also includes an annotation set list 376. Annotation set list 376 includes a listing of the various sets that correspond to the target media stream. According to one implementation, only those sets for which an annotation has been created are displayed in set list 376. According to one embodiment of the invention, **annotation server 10 of FIG. 3**

supports read and write security, allowing the creator of the set to identify which users are able to read and/or write to the annotation set. In this embodiment, only those sets for which the user has read access are displayed in set list 376.

Id., ¶[0109] (emphasis added).

Author field 182 contains data **identifying the user** who created annotation entry 180 and who is therefore the **author of the annotation**. The author is identified by ABE 151 of FIG. 3 based on the **user logged into client 15** at the time the annotation is created.

Id., ¶[0052] (emphasis added); *see also* ¶[0046] and FIG. 1 (teaching the access infrastructure).

132. In order for Gupta's computer system to provide such read and write access to particular users, the computer system includes functionality for receiving some form of authentication information (whether a username/password, IP address, public/private key, biometrics, etc.) so as to authenticate the user in question and determine whether this particular user has read and/or write access for a particular annotation. As noted, authentication information encompasses automatically generated information (*e.g.*, IP address of the requesting computer or a public or private key) and manually entered information (*e.g.*, username/password, biometrics). These methods of user authentication were common practice in networking at that time, and a POSITA would have understood that Gupta is

referring to such user authentication when it references terminology such as “access rights,” “read and write security,” and “write access.”

133. This authentication information is received over the network (*e.g.*, Internet), as Gupta describes that “the creator of the set” (*e.g.*, client 15) provides such “read and write controls” to annotation server 10 and such intercommunication is over public network 16.

... According to one embodiment of the invention, **annotation server 10** of FIG. 3 supports **read and write access controls**, allowing **the creator of the set to identify** which users are able to read and/or write to the annotation set. In this embodiment, only those sets for which the user has write access can be entered as set identifier 282. ...

Gupta, ¶[0090] (emphasis added); *see also* ¶¶[0109] and [0052] (already reproduced and emphasized above); ¶[0046] and FIG. 1.

F. Claim 19

134. Claim 19 recites:

19. The method of claim 18, further comprising denying or allowing access to all or part of the video editing environment based at least in part on the access authority of the remote client terminal.

135. In my opinion, Claim 19 is anticipated by Gupta.

136. Gupta discloses denying or allowing access to all or part of the video editing environment based at least in part on the access authority of the remote client terminal, as “**only those [annotation] sets** for which the user has write access can be entered as set identifier 282” for a user to add a new annotation. Therefore, Gupta discloses denying access, for the client 15, to parts of the software executed by servers 10-13 and stores 17-18.

... According to one embodiment of the invention, annotation server 10 of FIG. 3 supports read and write access controls, allowing the creator of the set to **identify which users are able to read and/or write to the annotation set**. In this embodiment, **only those sets for which the user has write access can be entered as set identifier 282**. Alternatively, rather than (or in addition to) user selection of an annotation set, one or more annotation sets may be inherently associated with dialog box 280 (or toolbar 250 of FIG. 6) and thus implicitly selected for each new annotation.

Gupta, ¶[0090] (emphasis added); *see also* ¶¶[0007], [0109].

137. I also refer to my discussion above of Claim 18.

G. Claim 20

138. Claim 20 recites:

The method of claim 13, wherein the server video data is streamed progressively to a video playback environment at the remote client terminal, allowing the video player to begin

displaying streamed server video data while remaining server video data continues to be streamed.

139. In my opinion, Claim 20 is anticipated by Gupta.

140. Gupta discloses the server video data being streamed progressively to a video playback environment (*e.g.*, media portion 612 including media screen 626) at the remote client terminal (*e.g.*, client 15), allowing the video player (*e.g.*, media screen 626) to begin displaying streamed server video data while remaining server video data continues to be streamed (*i.e.* client 15 renders streamed video data as it is received).

The term **“streaming”** is used to indicate that the data representing the various media types is **provided over a network to a client computer on a real-time, as-needed basis, rather than being pre-delivered in its entirety before playback**. Thus, the client computer **renders** streaming data **as it is received** from a network server, rather than waiting for an entire “file” to be delivered.

Gupta, ¶[0006] (emphasis added).

... Upon receipt of the segment specification from interface module 152, the browser communicates with media server 11 and requests the identified media segment using conventional HTTP requests. In response, **media server 11 streams the media segment to client 15 for presentation to the user**. This

presentation allows, for example, the user to verify the portion of the media stream to which his or her annotation will correspond.

Id., ¶[0099]; *see also* ¶[0146] (“Media portion 612 includes a media screen 626” which “is the area within which visual content of the multimedia content is rendered”), FIGs. 22-23.

A POSITA would have understood the word “progressively” (in “streamed progressively”) to mean streaming that allows display of the streamed server video data to begin while the remaining server video data is still being streamed, based on both the claim language and the ’286 Patent. *See* claim 20 (“streaming progressively ... allowing the video player to begin displaying streamed server video data while remaining server video data continues to be streamed.”); ’286 Patent 25:41-47 (“Further, video files may load in various ways, including as a complete file, streaming, or progressive. Progressive loading of video files, for example, allows the video frame to begin playing the portion of video data which has already loaded while the remainder of video data continues to load.”), 4:7-10 (“client video data is transmitted progressively, allowing the video player to begin displaying transmitted client video data while remaining client video data continues to be transmitted.”).

XI. GROUND 2: CLAIMS 13-15 AND 18-20 ARE ANTICIPATED BY FEGAN

A. Summary of Fegan

141. Fegan (Exhibit 1005) is a publication of a U.S patent application, published on June 30, 2005. I understand that Fegan constitutes prior art at least under pre-AIA 35 U.S.C. §§ 102(a) and (e) even under the alleged 2006 priority date.

142. Fegan describes a “comprehensive server-side system for editing, annotating, managing and presenting multimedia events.” Fegan, Abstract. A client accesses Fegan’s server-side system over the Internet to “generat[e] time-based index points” “for virtual segmentation of a multimedia file” (*e.g.*, video or streaming video) “allowing a user to create ‘clips’ of specified portions” that can be used to stream the selected portion. *Id.*, ¶¶[0080]-[0085], [0007] (“The application then uses the url/uri and time offsets to play the portion of the digital object that the user has selected”), [0013], [0076].

B. Claim 13

143. Claim 13 recites:

[13.P] 13. A method for handling video cue points, the method comprising:

[13.1] providing a computer system having at least one computer-readable medium comprising a video editing

environment, and being disposed at a server side of a public network;

[13.2] receiving a set of video information at the video editing environment using the computer system;

[13.3] receiving a set of cue point information at the video editing environment using the computer system;

[13.4] receiving a cue point editing command over the network from a remote client terminal disposed at a client side of the public network using the computer system;

[13.5] modifying a portion of the cue point information at the server side of the public network according to the cue point editing command received over the public network from the client terminal;

[13.6] receiving a cue point selection over the network from the remote client terminal identifying one of the cue points corresponding to a portion of the video information of one of the video files; and

[13.7] streaming the portion of the video information to the client terminal over the public network in response to receiving the cue point selection.

144. In my opinion, Claim 13 is anticipated by Fegan.

1. [13.P] A method for handling video cue points, the method comprising:

145. In my opinion, to the extent the preamble is limiting, Fegan discloses limitation [13.P].

146. Fegan discloses a method for handling (*e.g.*, “editing, annotating, managing”) video cue points (*e.g.*, “annotations”) for multimedia content such as videos.

The present invention relates to systems and methods for identifying, segmenting, collecting, annotating, and publishing multimedia materials. In some embodiments, the present invention provides a comprehensive server-side system for **editing, annotating, managing** and presenting multimedia events from a plurality of multimedia sources.

Fegan, Abstract (emphasis added).

The present invention is not limited by the nature of the multimedia files. For example, **multimedia files include**, but are not limited to, **streaming media**, audio information, **video information**, image information, and text information.

Id., ¶[0013] (emphasis added).

D. Segmenting Multimedia Content

As shown in FIG. 4, a user-selected multimedia file and compatible multimedia player are directed by software to be displayed in a browser window. In preferred embodiments, the

browser window contains additional tools for segmenting multimedia files. In particularly preferred embodiments, **the segmenting tools allow a user to generate time-based index points for a multimedia file** (e.g., spatial parameters for image files). The use of time-based/spatial-based index points provides a means for the virtual segmentation of a multimedia file, allowing a user to create “clips” of specified portions. Playback of the multimedia file may be started or stopped at any point in the file's timeline. The data associated with the time-based index points may be saved and stored on a remote server, and may be retrieved from the remote server and used whenever the associated multimedia file is accessed. Because the systems and methods of the present invention utilize time-based/spatial-based index points, multimedia editing may be achieved without the need to permanently store multimedia files, or divide them into separate files corresponding to each selected “clip.” The only data required to be stored for the purpose of segmenting multimedia files is the data associated with the time-based index points. The benefits of this approach include vastly decreased storage needs (multimedia files are often very large), faster access speeds over a computer network, and diminished copyright implications.

E. Annotating Multimedia Content

The systems and methods of the present invention provide a means for a user to add annotative information to a multimedia file. In preferred embodiments, such user-created **annotative**

information includes a title, annotative notes, and a designated location for storage and retrieval of the annotative information. FIG. 4 shows a browser window for editing of multimedia content that includes a multimedia file loaded into a compatible multimedia player, **controls for generating time-based index points for virtual segmentation of the multimedia file**, and a means of entering annotative information. In preferred embodiments, such annotative information is entered via one or more text fields in a browser window. **Annotative information typically comprises text information, although any type of multimedia data may be used (e.g., audio, video, image, etc.).** User-generated annotations may be used for any annotative purpose, including, but not limited to, providing contextual information, critical commentary, or descriptive information about the associated multimedia file.

After a user designates a location for storage and retrieval of the annotative information, the information appears on the user's personal portal page that can be accessed through a browser window. FIG. 5 shows a browser window displaying a user-created directory of annotative information associated with multimedia files. In preferred embodiments of the present invention, **the directory allows a user to select, open, edit, close, or delete the annotative information associated with a multimedia file**. FIG. 6 shows a browser window in which saved annotative information has been selected from the directory shown in FIG. 5. When a user selects an annotation, data

comprising the annotative information is displayed in the browser window, allowing the user to determine the contents of an annotation prior to opening it for editing. FIG. 7 shows a saved annotation that has been opened via the directory for further editing. **In preferred embodiments, annotations may be freely created, edited, and deleted without limitation.**

Id., ¶¶[0077]-[0081] (emphasis added).

147. Fegan describes “a comprehensive server-side system for editing, annotating, managing and presenting multimedia events from a plurality of multimedia sources.” *Id.*, Abstract.

The present invention relates to systems and methods for identifying, segmenting, collecting, annotating, and publishing multimedia materials. In some embodiments, the present invention provides a **comprehensive server-side system for editing, annotating, managing and presenting multimedia events from a plurality of multimedia sources.**

Id., Abstract (emphasis added).

148. Fegan discloses that its media segmentation and annotation publishing system “is maintained entirely on the server-side” and receive multimedia file annotation information from the client computer and combines those multimedia files and annotations into online publications, such that the user uses an online portal

to “create complex publications and presentations that include multimedia clips and annotations.”

In preferred embodiments, the **system is maintained entirely on the server-side**. The server (host) system manages the appropriate databases (e.g., user information, multimedia file location information, annotations, start and stop points, etc.). ...

Id., ¶[0010] (emphasis added).

Thus, in some embodiments, the present invention provides a **system for manipulating, annotating, and managing a plurality of remote, multimedia files in a server-side environment**, the system comprising a host computer network configured to carry out one or more of the following: a) identify multimedia files on an Internet web site, b) present the multimedia files to a client computer, c) receive playlist selection information (i.e., information pertaining to the identity of the multimedia files to be selected, annotated, or otherwise designated or manipulated) from the client computer, said playlist selection information comprising multimedia file identity and multimedia file start and stop points (i.e., designations of the starting and ending points of the multimedia file to be played in a presentation), d) **receive multimedia file annotation information from the client computer**, e) catalog (i.e., organize in a database or similar format) playlist selection information and multimedia file annotation information from a plurality of web sites selected by the client computer, and f)

combine those multimedia files and annotations into online publications.

Id., ¶[0012] (emphasis added).

In preferred embodiments, the systems of the present invention are made available to client computers in an online environment. Use of the systems of the present invention online creates flexibility and ease-of-use for developing, managing, and using presentations that contain multimedia information. For example, the systems of the **present invention allow users to find, segment, annotate, organize, and publish streaming media on the World Wide Web.** In preferred embodiments, information is accessed by the user through the use of convenient bookmarks and hyperlinks. For example, a user's browser's bookmark feature is used to launch the application. ... In some embodiments, each user has a **personal "portal" hosted on the system** of the present invention that allows the user to edit and organize collected materials and to **create complex publications and presentations that include multimedia clips and annotations.**

Id., ¶[0009] (emphasis added).

149. Fegan touts its system as advantageously providing "the ability to manage complex presentations without burdening the client's computer hardware and allowing access to the presentation from any number of client computers." *Id.*, ¶[0008]. Therefore, Fegan's annotations constitute the claimed "cue points," which

the '286 Patent describes as “indexed time positions or tags anywhere within a video file” and may include “meta-tags” that “describe the information in the video file” and “include descriptive identifiers.” '286 Patent, 7:50-52, 8:15-20.

150. Therefore, Fegan’s annotations are “cue points” in the context of the '286 Patent claims.

2. [13.1] providing a computer system having at least one computer-readable medium comprising a video editing environment, and being disposed at a server side of a public network;

151. In my opinion, Fegan discloses limitation [13.1].

152. Fegan discloses providing a computer system including servers and storage such as databases at a server side of a public network. The computer system includes at least one remote server connected to a network, because Fegan’s system is “hosted in an entirely server-side environment” where “the identity of the files, the annotations, and the organization of the files is stored and managed on the host network.”

Thus, in addressing the limitations of the prior art, the present invention provides comprehensive systems for identifying, segmenting, collecting, annotating, and publishing multimedia materials. In preferred embodiments, the **systems and methods are hosted in an entirely server-side environment**, such that client computers can identify, segment, collect, annotate, organize, and publish complex multimedia presentations

containing multimedia files or portions of multimedia files from a plurality of sources (e.g., any web site offering multimedia on the internet). ...

Fegan, ¶[0008] (emphasis added).

In preferred embodiments, the system is **maintained entirely on the server-side**. The **server (host) system manages** the appropriate **databases** (e.g., **user information, multimedia file location information, annotations, start and stop points, etc.**).

...

Id., ¶[0010] (emphasis added).

153. Further, Fegan discloses its network is a “public” network (*e.g.*, “Internet”). Fegan also explains that multimedia presentations, such as videos, are “stored entirely on a remote server.”

In some embodiments of the present invention, the functionality of the browser is extended by means of one or more separate computer programs (hereinafter referred to as “applications”) that are stored in one or more remote locations (e.g., locations other than locally on the user’s computer). In preferred embodiments, functionality for locating multimedia content is added to a browser via one or more hyperlinks that launch the remote applications. FIG. 1 shows an example of multimedia content search functionality that can be added to a browser by means of a hyperlink. The hyperlink may be stored by the browser in a manner that renders it easily accessible, such as on

a browser's toolbar or in a list of bookmarks. FIG. 2 shows a browser window in which a hyperlink adding multimedia content search functionality has been added to the browser's toolbar for convenient access. The hyperlink connects the user to a **remote server** via a **computer network (e.g., the Internet)**. ...

Id., ¶[0071] (emphasis added).

The systems and methods of the present invention may be used to publish stored multimedia presentations by making them available for access by others over a **computer network (e.g., the Internet)**. For example, a first user in a first location may create a multimedia presentation that is **stored entirely on a remote server**. ...

Id., ¶[0085] (emphasis added).

154. Fegan's computer system (including servers and storage) has at least one computer-readable medium (*e.g.*, server memory and/or storage) because Fegan's system runs on "server-side software" such as MYSQL and PHP, which uses the remote server's memory/storage.

In preferred embodiments, the system is maintained entirely on the **server-side**. The server (host) system manages the appropriate databases (*e.g.*, user information, multimedia file location information, annotations, start and stop points, etc.). The present invention is not limited to particular types or kinds of server-side software. In some preferred embodiments, the databases are driven by **MYSQL**, freely available database

software for Unix and Win32/WinNT (see, e.g., MYSQL, Second Edition, by Paul DuBois, Pearson Education, 2003, herein incorporated by reference in its entirety). In some preferred embodiments, the features of the present invention are encoded in **PHP** and JavaScript, with an XML-based delivery and display (see, e.g., Advanced PHP Programming, George Schlossnagle, SAMS, 2003, herein incorporated by reference in its entirety). Software and tools for use in the present invention is described and available at MATRIX, Michigan State University (Lansing, Mich.) (matrix.msu.edu/innermatrix/about.php). One advantage of this system is the ability to publish and present multimedia events with no proliferation of the digital files encoding the multimedia events (i.e., embodiments of the present invention provide a pointer to the media rather than a copy of the media).

Id., ¶[0010] (emphasis added).

The selected multimedia file and compatible multimedia player are then displayed in a browser window. FIG. 4 shows a browser window in which a user-selected multimedia file and a compatible multimedia player are displayed. The window is configured to provide user access to the playback controls of the multimedia player. In preferred embodiments, the playback controls include the functions typical to most multimedia players, including, but not limited to, the ability to start, stop, and pause playback in real time, move in forward and reverse at speeds faster and slower than real time (e.g., fast-forward,

rewind, slow motion, etc.), and adjust playback conditions (e.g., audio volume, image brightness and contrast, etc.). In some embodiments, **software that is stored on a remote server** allows a user to save and store the desired playback parameters (e.g., audio volume) associated with a selected multimedia file. The stored playback parameters may be retrieved from the remote server and used whenever the associated multimedia file is accessed.

Id., ¶[0076] (emphasis added).

The systems and methods of the present invention may be used to create user-customizable multimedia presentations comprising multimedia files, segment information, and annotative information. In preferred embodiments, **software that is stored on a remote server** allows a user to save and store segment and annotative information associated with one or more multimedia files. Because the systems and methods of the present invention utilize time and/or spatial-based index points and annotations, multimedia presentations may be created without the need to permanently store multimedia files, or divide them into separate files corresponding to each selected “clip.” In some embodiments, **software that is stored on a remote server** provides one or more templates upon which a multimedia presentation may be based. In preferred embodiments, both preset and customizable templates are provided. In other embodiments, the multimedia presentation is executed by means of a specialized programming language. In particularly preferred

embodiments, the multimedia presentation is executed by means of Synchronized Multimedia Integration Language (“SMIL”). In yet other embodiments, the multimedia presentation is executed by means of Hypertext Markup Language (“HTML”), although any suitable programming language may be used.

Id., ¶[0083] (emphasis added).

As used herein, the terms "processor" and "central processing unit" or "CPU" are used interchangeably and refer to a device that is able to **read a program from a computer memory (e.g., ROM or other computer memory)** and perform a set of steps according to the program.

As used herein, the terms "computer memory" and "computer memory device" refer to any storage media readable by a computer processor. Examples of computer memory include, but are not limited to, **RAM, ROM, computer chips, digital video discs (DVD), compact discs (CDs), hard disk drives (HDD), and magnetic tape.**

As used herein, the term "computer readable medium" refers to **any device or system for storing and providing information (e.g., data and instructions) to a computer processor.** Examples of computer readable media include, but are not limited to, DVDs, CDs, hard disk drives, magnetic tape and servers for streaming media over networks.

Id., ¶¶[0024]-[0026] (emphasis added).

As used herein, the term "client-server" refers to a model of interaction in a distributed system in which a program at one site sends a request to a **program at another site** and waits for a response. The requesting program is called the "client," and the **program that responds to the request is called the "server."** In the context of the World Wide Web (discussed below), the client is a "Web browser" (or simply "browser") that runs on a computer (e.g., desktop, cell phone, hand-held, etc.) of a user; the **program which responds to browser requests by serving Web pages is commonly referred to as a "Web server."**

Id., ¶[0037] (emphasis added).

155. Fegan describes the at least one computer readable medium comprising a video editing environment, which includes software for both editing annotations and streaming video executed by the remote servers and databases. For example, Fegan describes that the software for editing annotations and streaming media is stored in one or more "remote locations (e.g. locations other than locally on the user's computer)" such as a remote server.

In some embodiments of the present invention, the functionality of the browser is extended by means of one or more separate **computer programs** (hereinafter referred to as "applications") that are **stored in one or more remote locations (e.g., locations other than locally on the user's computer)**. In preferred embodiments, functionality for locating multimedia content is added to a browser via one or more hyperlinks that launch the

remote applications. FIG. 1 shows an example of multimedia content search functionality that can be added to a browser by means of a hyperlink. The hyperlink may be stored by the browser in a manner that renders it easily accessible, such as on a browser's toolbar or in a list of bookmarks. FIG. 2 shows a browser window in which a hyperlink adding multimedia content search functionality has been added to the browser's toolbar for convenient access. The hyperlink connects the user to a remote server via a computer network (e.g., the Internet). Software (e.g., one or more applets) that is stored on the remote server allows any location on a computer network to be automatically searched for multimedia content simply by accessing the hyperlink from that location. In preferred embodiments, the location to be searched is an Internet web page, although any accessible location on a computer network may be searched for multimedia content.

Id., ¶[0071] (emphasis added).

After a location on a computer network is searched, the results of the search are displayed in a window. Any multimedia content found at the searched location can be displayed. The display can be arranged by media type (e.g., video, audio, image, text, etc.), file type, file name, location (e.g., URL), or by any other desired means. FIG. 3 shows a browser window in which the results of a search for multimedia content on a web page are displayed by location. In preferred embodiments of the present invention, the data comprising the search results may be stored by **software on**

a remote server. The stored data includes information useful to locate, identify, and categorize the multimedia files, but preferably excludes the multimedia files themselves. As shown in FIG. 3, a means is provided for a user to select one or more of the identified multimedia files for access.

After a desired multimedia file has been selected, software directs the browser to load the file into a browser-based multimedia player that is compatible with the file type of the selected multimedia file. Examples of browser-based multimedia players that may be used with the present invention include RealPlayer (Real Networks), Windows Media Player (Microsoft), and QuickTime Player (Apple), although any browser-based multimedia player may be used.

Id., ¶¶[0073]-[0074] (emphasis added).

The systems and methods of the present invention may be used to create user-customizable multimedia presentations comprising multimedia files, segment information, and annotative information. In preferred embodiments, software that is stored on a remote server allows a user to save and store segment and annotative information associated with one or more multimedia files. Because the systems and methods of the present invention utilize time and/or spatial-based index points and annotations, multimedia presentations may be created without the need to permanently store multimedia files, or divide them into separate files corresponding to each selected “clip.” In some embodiments, software that is stored on a remote server

provides one or more templates upon which a multimedia presentation may be based. In preferred embodiments, both preset and customizable templates are provided. In other embodiments, the multimedia presentation is executed by means of a specialized programming language. In particularly preferred embodiments, the multimedia presentation is executed by means of Synchronized Multimedia Integration Language (“SMIL”). In yet other embodiments, the multimedia presentation is executed by means of Hypertext Markup Language (“HTML”), although any suitable programming language may be used.

Id., ¶[0083] (emphasis added).

... **The technical tools for carrying out the multimedia presentation are maintained on the server-side**, rather than forcing the user on the client-side to navigate the difficulties of the management of the multimedia files.

Id., ¶[0007] (emphasis added);

In preferred embodiments, the systems of the present invention are made available to client computers in an online environment. Use of the systems of the present invention online creates flexibility and ease-of-use for developing, managing, and using presentations that contain multimedia information. ... In some embodiments, **each user has a personal "portal" hosted on the system** of the present invention that allows the user to edit and organize collected materials and to create complex publications and presentations that include multimedia clips and annotations.

Id., ¶[0009] (emphasis added); *see also* FIGs. 1-7.

156. Fegan's computer system is disposed at a server side of a public network (such as the Internet). By contrast, Fegan's client computers are disposed at a client side of the network.

Thus, in addressing the limitations of the prior art, the present invention provides comprehensive systems for identifying, segmenting, collecting, annotating, and publishing multimedia materials. In preferred embodiments, the **systems and methods are hosted in an entirely server-side environment**, such that client computers can identify, segment, collect, annotate, organize, and publish complex multimedia presentations containing multimedia files or portions of multimedia files from a plurality of sources (e.g., any web site offering multimedia on the internet). ... An advantage of such systems is the ability to manage complex presentations **without burdening the client's computer hardware** and allowing access to the presentation from any number of client computers. Specific illustrative, non-limiting embodiments of the present invention are described below in the Detailed Description of the Invention.

Id., ¶[0008] (emphasis added).

In some embodiments of the present invention, the functionality of the browser is extended by means of one or more separate **computer programs** (hereinafter referred to as "applications") that are stored in one or more **remote locations (e.g., locations**

other than locally on the user's computer). ... FIG. 2 shows a browser window in which a hyperlink adding multimedia content search functionality has been added to the browser's toolbar for convenient access. The hyperlink connects the user to a remote server via a computer network (e.g., the Internet). ...

Id., ¶[0071] (emphasis added).

157. Notably, the '286 Patent broadly describes that a “video editing environment” (or “VEE”) “may comprise any graphical, textual, audio, or other data which is desirable or necessary for providing interface capabilities to a client” and “does not [even] require any graphical interface elements” (*e.g.*, even encompasses “script”-level interfaces). '286 Patent, 13:48-67.

3. [13.2] receiving a set of video information at the video editing environment using the computer system;

158. In my opinion, Fegan discloses limitation [13.2].

159. Fegan discloses multiple instances where the computer system (*e.g.*, software for streaming video using the remote servers and databases) receives a set of video information (*e.g.*, a collection of information concerning video).

160. A first instance of Fegan's computer system receiving a set of video information is Fegan's “remote server” receiving and storing videos because “a first user in a first location may create a multimedia presentation that is stored entirely on a remote server.” The remote server therefore must receive the video information

(e.g., videos and streaming media) and store it in order to enable the multimedia presentation to be “stored entirely on a remote server.”

The systems and methods of the present invention may be used to publish stored multimedia presentations by making them available for access by others over a computer network (e.g., the Internet). For example, a first user in a first location may create a multimedia presentation that is **stored entirely on a remote server**. In preferred embodiments of the present invention, one or more additional users in one or more separate and distinct locations on a computer network may access the multimedia presentation created by the first user. ... In addition, **the server-side approach of the present invention eliminates the necessity of storing any of the data associated with the multimedia presentation on a user’s local computer**.

Fegan, ¶[0085] (emphasis added).

161. A second instance of Fegan’s computer system receiving a set of video information is where Fegan’s remote server receives information on videos stored at other locations, because Fegan describes searching locations at “any available location” on the network.

... Software (e.g., one or more applets) that is stored on the remote server allows **any location on a computer network to be automatically searched for multimedia content simply by accessing the hyperlink from that location**. In preferred embodiments, the location to be searched is an Internet web page,

although **any accessible location on a computer network may be searched for multimedia content.**

Id., ¶[0071] (emphasis added).

... FIG. 3 shows a browser window in which the results of a search for multimedia content on a web page are displayed by location. In preferred embodiments of the present invention, the **data comprising the search results may be stored by software on a remote server.** The stored data includes information useful to locate, identify, and categorize the multimedia files, but preferably excludes the multimedia files themselves. ...

Id., ¶[0073] (emphasis added).

This would include videos stored at the “remote server” (*id.*, ¶[0085]) or even videos stored at other locations on the network including webpages connected to the remote server via the Internet (*id.*, ¶[0069]-[0071]). Thus, Fegan discloses that video information (*e.g.*, videos and streaming media) is received by Fegan’s computer system when the webpages/servers connected to the remote server via the Internet receive videos or streaming media for storage. This is consistent with the ’286 Patent’s description of video servers being part of the computer system by virtue of being on the same network as the other servers that make up the computer system.

In one example, the video server 112 may be a ... network of computers with the capability to store and index mass amounts

of video data ... to access a video file 113, a command may be sent to the video server 112 with an identifier ... of a video file

'286 Patent, 7:15-19.

In another example, the video server may comprise a list stored in one or more files which matches video identifiers with the location of respective videos. The videos may be located in the same data storage device as the video server, or in Some other location, Such as at some internet address, file transfer protocol (FTP) site, or some other protocol location (e.g. by hyper-text transfer protocol, HTTP).

'286 Patent, 7:23-26; *see also* 13:45-57.

162. A third instance of Fegan's computer system receiving a set of video information is where Fegan's computer system (*e.g.*, remote server) receives and stores URL location information of videos. For example, Fegan describes searching a location (*e.g.*, webpages/servers connected to the remote server via the Internet) for media and displaying the results of the search to a user, including the video URLs. Fegan's remote server must first receive the video information (*e.g.*, URLs) in order to store them.

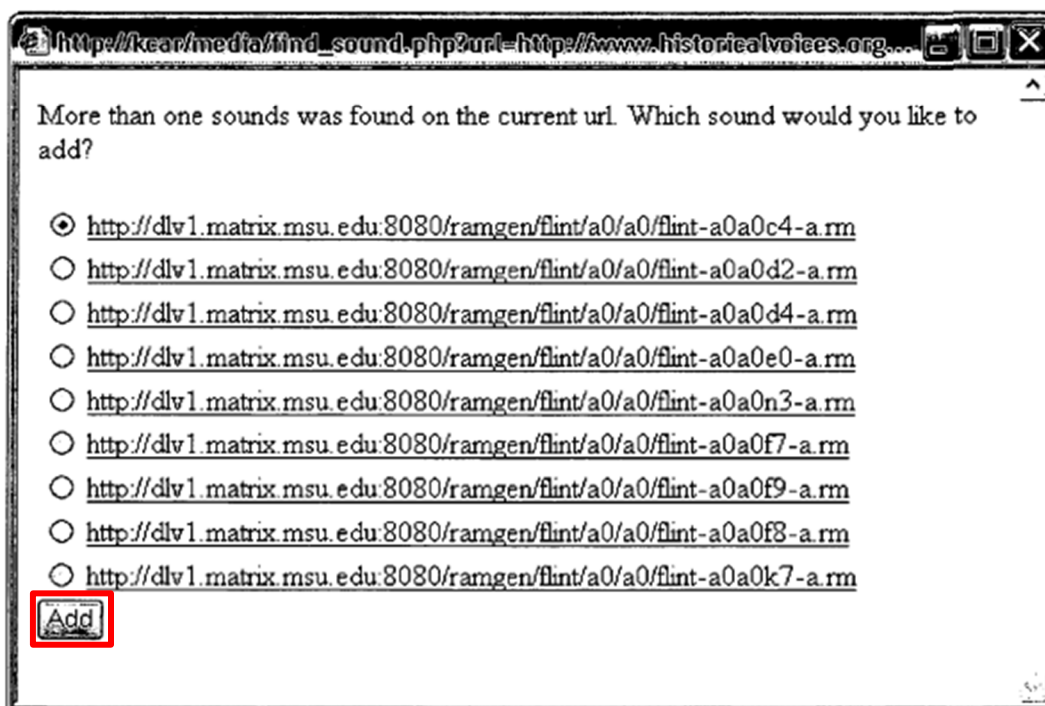
After a location on a computer network is searched, the results of the search are displayed in a window. Any multimedia content found at the searched location can be displayed. The display can be arranged by media type (*e.g.*, video, audio, image, text, etc.), file type, file name, location (*e.g.*, URL), or by any other desired

means. FIG. 3 shows a browser window in which the results of a search for multimedia content on a web page are displayed by location. In preferred embodiments of the present invention, the data comprising the search results may be stored by software on a remote server. The stored data includes information useful to locate, identify, and categorize the multimedia files, but preferably excludes the multimedia files themselves. As shown in FIG. 3, a means is provided for a user to select one or more of the identified multimedia files for access.

Fegan, ¶[0073] (emphasis added).

FIGURE 3

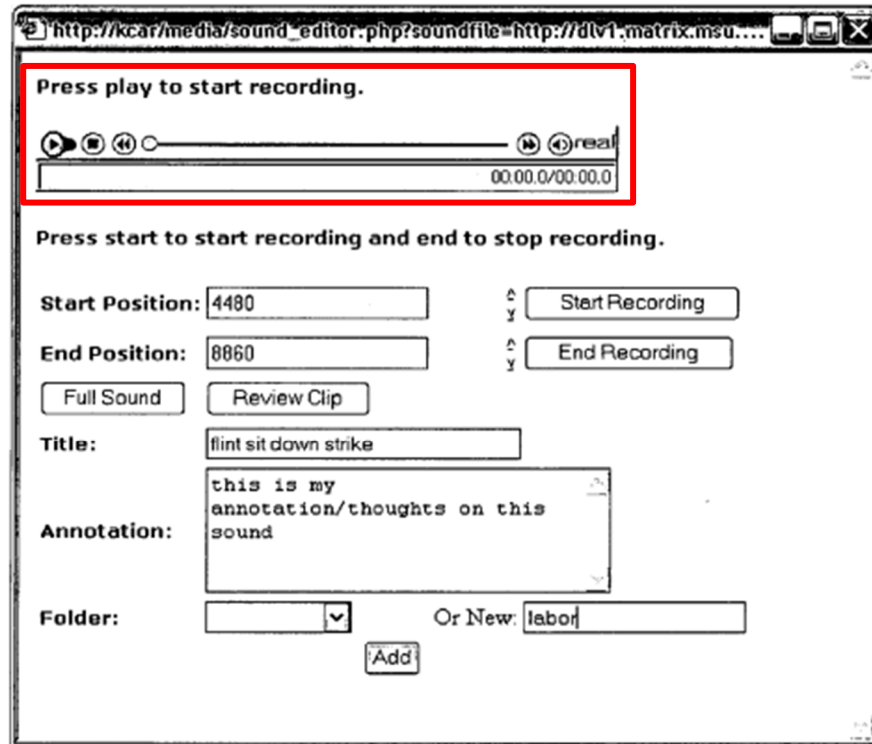
Go to any web page and click the "Search for Sound" bookmark. If the page contains audio, it will load it into the editor. If it contains more than sound, it will list all the sounds and allow you to select the one you want to edit (shown below). If there are no sounds, it will simply tell you that no sounds are available.



Id., Fig. 3 (annotated).

FIGURE 4

The system will load the sound into the editor and allow you to create a segment by pressing start and end recording and then adding a title, annotation, and dictating what folder this annotation should go in on your portal page.



Id., Fig. 4 (annotated).

163. The set of video information is received at the video editing environment using the computer system in Fegan, because videos and video URLs are received by the software executed on the remote server and by accessible locations on the remote server's network, which are servers within Fegan's system used to stream video.

The present invention relates to systems and methods for identifying, segmenting, collecting, annotating, and publishing multimedia materials. In some embodiments, the present

invention provides a comprehensive server-side system for editing, annotating, managing and presenting multimedia events from a plurality of multimedia sources.

Id., ¶[0003] (emphasis added).

The present invention relates to systems and methods for identifying, segmenting, collecting, annotating, and publishing multimedia materials that are stored on remote servers. In some embodiments, the present invention provides a comprehensive server-side system for managing and presenting multimedia events from a plurality of multimedia sources.

Id., ¶[0005] (emphasis added).

In preferred embodiments, the system is maintained entirely on the server-side. The server (host) system manages the appropriate databases (e.g., user information, multimedia file location information, annotations, start and stop points, etc.). ...

Id., ¶[0010] (emphasis added).

After a desired multimedia file has been selected, software directs the browser to load the file into a browser-based multimedia player that is compatible with the file type of the selected multimedia file. Examples of browser-based multimedia players that may be used with the present invention include RealPlayer (Real Networks), Windows Media Player (Microsoft), and QuickTime Player (Apple), although any browser-based multimedia player may be used.

Id., ¶[0074] (emphasis added); *see also* ¶¶[0069]-[0073], [0085].

In preferred embodiments, the systems of the present invention are made available to client computers in an online environment. Use of the systems of the present invention online creates flexibility and ease-of-use for developing, managing, and using presentations that contain multimedia information. For example, the systems of the present invention allow users to find, segment, annotate, organize, and publish streaming media on the World Wide Web. In preferred embodiments, information is accessed by the user through the use of convenient bookmarks and hyperlinks. For example, a user's browser's bookmark feature is used to launch the application. In some preferred embodiments, the system of the present invention provides ease-of-use in assembling a presentation. For example, in some embodiments, the system selects an appropriate multimedia player (e.g., with selection based on the nature of the content to be played) and finds and loads the desired media into an editor. **The media is displayed to the user along with annotation tools that allow the user to isolate portions of the media relevant to the presentation and to annotate the media as desired.** In some embodiments, each user has a personal "portal" hosted on the system of the present invention that allows the user to edit and organize collected materials and to create complex publications and presentations that include multimedia clips and annotations.

Id., ¶[0009] (emphasis added).

164. Thus the computer system (*e.g.*, remote server) receives a set of video information by searching for media, receiving the results of the search as the location of the media (*e.g.*, URLs), and then storing these locations. This is consistent with the '286 Patent's description of "**the video server may comprise a list** stored in one or more files which matches video identifiers **with the location of respective videos** ... In this example, to access a video file 113, a command may be sent to the video server 112 with a video file identifier ['VID-0001'] ... a relational database may be queried to find an entry for 'VID—0001'. This entry may be stored in relation to **a list of URL and FTP destinations.**" '286 Patent, 7:22-39 (emphasis added).

4. [13.3] receiving a set of cue point information at the video editing environment using the computer system;

165. In my opinion, Fegan discloses limitation [13.3].

166. Fegan discloses receiving a set of cue point information (*e.g.*, a collection of information concerning annotations) at the video editing environment using the computer system (*e.g.*, at the software for editing annotations using the remote server and databases). For example, Fegan discloses that the annotations added by users are received by the remote servers and stored and managed on the host network, including in a server-side environment storing the multimedia file annotation information to generate a multimedia presentation, and Fegan also discloses providing the client computer access to the multimedia presentation.

Thus, in addressing the limitations of the prior art, the present invention provides comprehensive systems for identifying, segmenting, collecting, annotating, and publishing multimedia materials. In preferred embodiments, the systems and methods are hosted in an entirely server-side environment, such that client computers can identify, segment, collect, annotate, organize, and publish complex multimedia presentations containing multimedia files or portions of multimedia files from a plurality of sources (e.g., any web site offering multimedia on the internet). In some such preferred embodiments, the identity of the files, **The annotations**, and the organization of the files **is stored and managed on the host network**. In preferred embodiments, the multimedia files are maintained only on the third party web sites and are not copied onto the host or client computers. An advantage of such systems is the ability to manage complex presentations without burdening the client's computer hardware and allowing access to the presentation from any number of client computers. ...

Fegan, ¶[0008] (emphasis added).

In preferred embodiments, the system is maintained entirely on the server-side. The server (host) system **manages the appropriate databases** (e.g., user information, multimedia file location information, **annotations**, **start and stop points**, etc.)

Id., ¶[0010] (emphasis added).

The present invention also provides a method for managing a plurality of multimedia files in a server-side environment, comprising one or more of the following steps: a) providing the systems of the present invention; b) identifying multimedia files on an Internet web site selected by the client computer; c) displaying identified multimedia files to the client computer; d) receiving the playlist selection information from the client computer; e) receiving the multimedia file annotation information from the client computer; f) storing the playlist selection information and the multimedia file annotation information to generate a multimedia presentation; and g) providing the client computer access to the multimedia presentation.

Id., ¶[0015] (emphasis added).

Thus, in some embodiments, the present invention provides a system for manipulating, annotating, and managing a plurality of remote, multimedia files in a server-side environment, the system comprising a host computer network configured to carry out one or more of the following: a) identify multimedia files on an Internet web site, b) present the multimedia files to a client computer, c) receive playlist selection information (i.e., information pertaining to the identity of the multimedia files to be selected, annotated, or otherwise designated or manipulated) from the client computer, said playlist selection information comprising multimedia file identity and multimedia file start and stop points (i.e., designations of the starting and ending points of

the multimedia file to be played in a presentation), d) **receive multimedia file annotation information from the client computer**, e) catalog (i.e., organize in a database or similar format) playlist selection information and multimedia file annotation information from a plurality of web sites selected by the client computer, and f) combine those multimedia files and annotations into online publications.

Id., ¶[0012] (emphasis added); *see also* ¶[0078].

167. In Fegan, the set of cue point information is received at the video editing environment using the computer system because annotation and start and stop points are received via the interfaces and program modules of the remote server of Fegan's system used to edit annotations.

168. For example, FIG. 4 shows adding an annotation including a start and end position and title to a user's personal page on the server in a new folder labeled "labor" and FIG. 5 shows a browser window displaying a user-created directory of annotative information associated with multimedia files.

The present invention relates to systems and methods for identifying, segmenting, collecting, annotating, and publishing multimedia materials. In some embodiments, the present invention provides a comprehensive **server-side system for editing, annotating, managing and presenting multimedia events** from a plurality of multimedia sources.

Id., ¶[0003] (emphasis added).

The present invention also provides a method for managing a plurality of multimedia files in a server-side environment, comprising one or more of the following steps: a) providing the systems of the present invention; b) identifying multimedia files on an Internet web site selected by the client computer; c) displaying identified multimedia files to the client computer; d) receiving the playlist selection information from the client computer; e) receiving the multimedia file annotation information from the client computer; f) storing the playlist selection information and the multimedia file annotation information to generate a multimedia presentation; and g) providing the client computer access to the multimedia presentation.

Id., ¶[0015] (emphasis added).

As shown in FIG. 4, a user-selected multimedia file and compatible multimedia player are directed by software to be displayed in a browser window. In preferred embodiments, the browser window contains additional tools for segmenting multimedia files. In particularly preferred embodiments, the segmenting tools allow a user to generate time-based index points for a multimedia file (e.g., spatial parameters for image files). The use of time-based/spatial-based index points provides a means for the virtual segmentation of a multimedia file, allowing a user to create "clips" of specified portions. Playback of the multimedia file may be started or stopped at any point in the file's timeline. The data associated with the time-based

index points may be saved and stored on a remote server, and may be retrieved from the remote server and used whenever the associated multimedia file is accessed. Because the systems and methods of the present invention utilize time-based/spatial-based index points, multimedia editing may be achieved without the need to permanently store multimedia files, or divide them into separate files corresponding to each selected "clip." The only data required to be stored for the purpose of segmenting multimedia files is the data associated with the time-based index points. The benefits of this approach include vastly decreased storage needs (multimedia files are often very large), faster access speeds over a computer network, and diminished copyright implications.

Id., ¶[0078] (emphasis added).

... FIG. 4 shows a browser window for editing of multimedia content that includes a multimedia file loaded into a compatible multimedia player, controls for generating time-based index points for virtual segmentation of the multimedia file, and a means of entering annotative information. ...

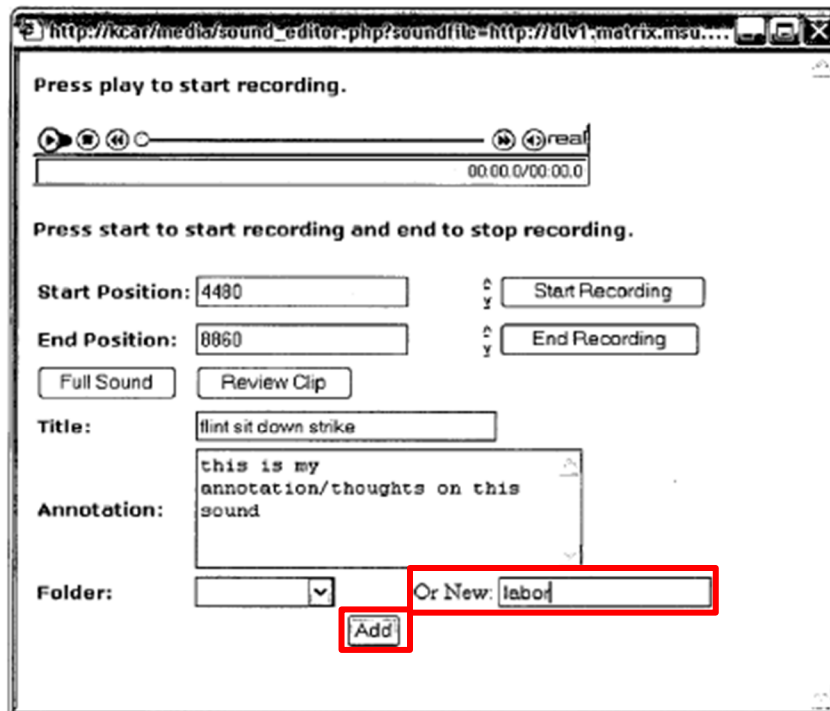
After a user designates a location for storage and retrieval of the annotative information, the information appears on the user's personal portal page that can be accessed through a browser window. **FIG. 5 shows a browser window displaying a user-created directory of annotative information associated with multimedia files.** In preferred embodiments of the present invention, the directory allows a user to select, open, edit, close,

or delete the annotative information associated with a multimedia file. ...

Id., ¶¶[0080]-[0081] (emphasis added); *see also* ¶[0083].

FIGURE 4

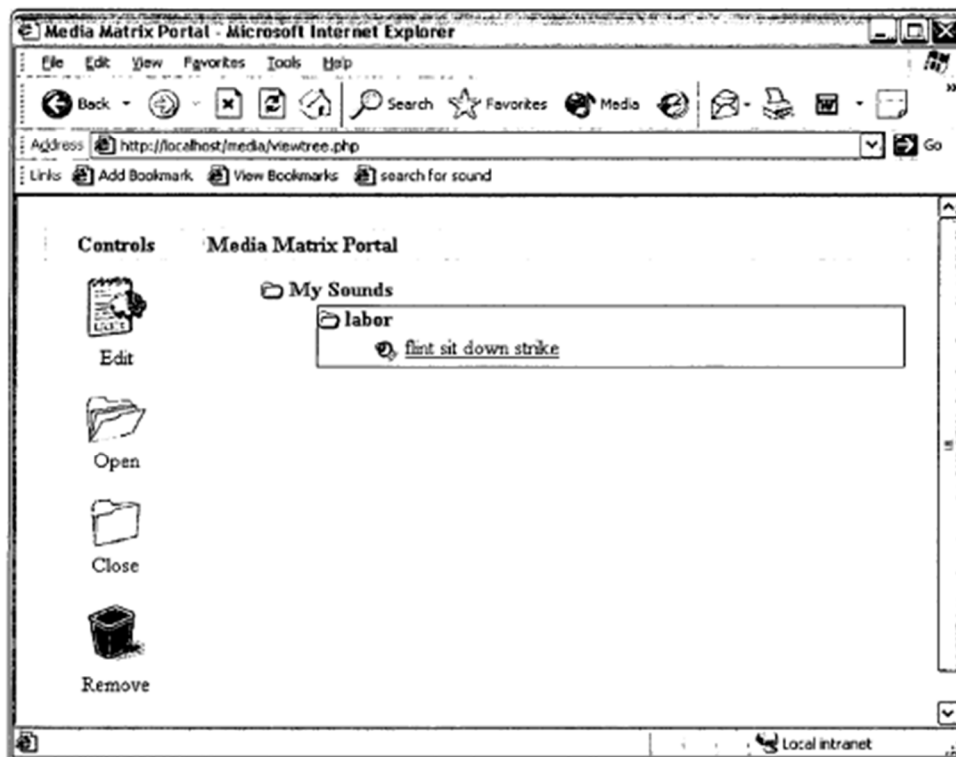
The system will load the sound into the editor and allow you to create a segment by pressing start and end recording and then adding a title, annotation, and dictating what folder this annotation should go in on your portal page.



Id., FIG. 4 (annotated, showing adding annotation to user's personal page on the server in a new folder labeled "labor").

FIGURE 5

The folder created that appears on the user's portal page along with the annotation. Clicking on the folder shows the annotations in the folder as well as bringing up the controls to edit the settings of the folder (rename it, move it to another folder, delete it, etc...)



Id., FIG. 5 (showing the added annotation “flint sit down strike” saved to the new “labor” folder).

5. [13.4] receiving a cue point editing command over the network from a remote client terminal disposed at a client side of the public network using the computer system;

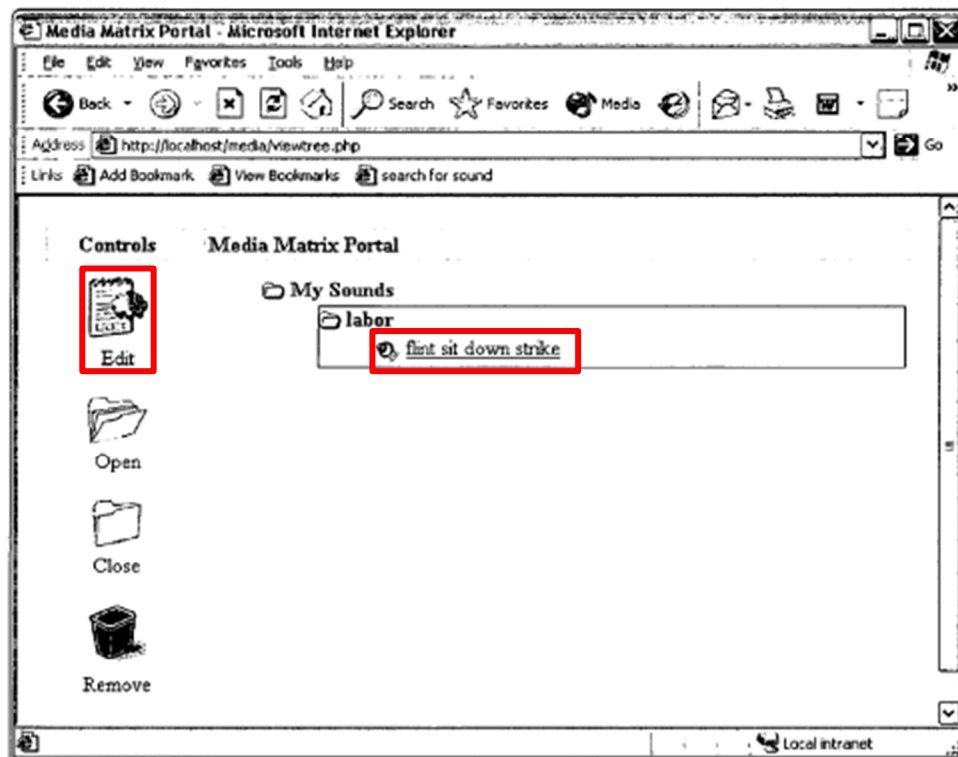
169. In my opinion, Fegan discloses limitation [13.4].

170. Fegan discloses the computer system receiving a cue point editing command (e.g., command to edit an annotation after a client selects an “update” button) from a remote client terminal. For example, Fegan describes that to edit an

existing annotation, a client user enters an edit mode by selecting an existing annotation (e.g., “flint sit down strike”) and selecting the left-side “Edit” control.

FIGURE 5

The folder created that appears on the user’s portal page along with the annotation. Clicking on the folder shows the annotations in the folder as well as bringing up the controls to edit the settings of the folder (rename it, move it to another folder, delete it, etc...)



Fegan, FIG. 5 (annotated).

When a user selects an annotation, data comprising the annotative information is displayed in the browser window, allowing the user to determine the contents of an annotation prior to opening it for editing.” After a user designates a location for storage and retrieval of the annotative information, the information appears on the user’s personal portal page that can

be accessed through a browser window. **FIG. 5** shows a browser window displaying a **user-created directory of annotative information** associated with multimedia files. In preferred embodiments of the present invention, **the directory allows a user to select, open, edit, close, or delete the annotative information associated with a multimedia file.** ...

Id., ¶[0081].

171. Fegan explains that when a user selects “Edit” for the annotation, the user enters an edit mode such as shown in FIG. 7 which shows a saved annotation that has been opened via the directory for further editing.

FIGURE 7
by clicking the edit button, a user can edit any of the information for that annotation including re-editing the time-offsets from the source.

Press play to start recording.

Press start to start recording and end to stop recording.

Start Position: 3791 Start Recording

End Position: 7474 End Recording

Full Sound Review Clip

Title: flint sit down strike

Annotation: my annotation/thoughts on this sound

Folder: -labor Or New: Update

Id., FIG. 7 (annotated, showing exemplary editable fields and “update” button for saving annotation edits to the server).

... **FIG. 7** shows a **saved annotation that has been opened via the directory for further editing**. In preferred embodiments, annotations may be freely created, edited, and deleted without limitation.

Id., ¶[0081] (emphasis added).

172. Fegan explains that “annotations may be freely created, edited, and deleted without limitation” (*id.*, ¶[0081]) and “software [] stored on a remote server allows a user to save and store segment and annotative information associated with one or more multimedia files” thus allowing creation of multimedia presentations “without the need to permanently store multimedia files, or divide them into separate files corresponding to each selected ‘clip’” (*id.*, ¶[0083]). *Id.*, ¶[0085] (“the server-side approach of the present invention eliminates the necessity of storing any of the data associated with the multimedia presentation on a user’s local computer”), [0081] (“the directory allows a user to select, open, edit, close, or delete the annotative information”). Thus, after editing text fields as shown in FIG. 7, when the user selects “Update,” a cue point editing command to edit the annotation is sent from the client computer to the server and is therefore received at the computer system’s remote server. *Id.*, ¶[0003] (“the present invention provides a comprehensive server-side system for editing, annotating, managing and presenting

multimedia events from a plurality of multimedia sources”), [0009] (“each user has a personal ‘portal’ hosted on the system” “that allows the user to edit ... presentations that include multimedia clips and annotations.”)

173. A POSITA would have understood the plain meaning of “cue point editing command” to be a command, sent from the remote client terminal and received by the computer system, to edit a portion of the cue point information. *See* limitation [13.5] (indicating that “according to the cue point editing command” “a portion of the cue point information” is “modif[ied]”) (emphasis added). Further, a POSITA would have understood the word “command” (in “cue point editing command”) to mean an instruction to a computer program, due to how this word is used in the ’286 Patent.

174. The ’286 Patent describes that “the command, ‘RETRIEVE (‘VID—0001’), is used to retrieve a video file identified as ‘VID—0001’ from the video server 112. When the video server 112 receives the ‘RETRIEVE’ command, a relational database may be queried to find an entry for ‘VID—0001’ ... The video server 112 may then use various algorithms to select a download location and download the file.” ’286 Patent, 7:33-41 (emphasis added). In other words, an instruction is sent to the video server to retrieve a video, to which the video server responds. *See also id.*, 13:65-67 (describing “Visual Basic commands for an interpreter” that are instructions to the interpreter “to generate the GUI”) (emphasis

added), 42:45-48 (describing a “**command** 1640” “transmitted by a client, received by a host or application” “requesting that a dataset is built for all ‘Exterior’ cue points” and the host or application responds to the instruction by “generat[ing] dataset 1650.”) (emphasis added).

175. This understanding, that a “command” is an instruction to a computer program, also comports with contemporaneous technical dictionaries, *e.g.*, the Microsoft Computer Dictionary defines a “command” as “An instruction to a computer program that, when issued by the user, causes an action to be carried out.” Exhibit 1017.

176. Therefore, a POSITA would have understood that the user’s selection of “update” after changing annotation information in the FIG. 7 interface results in sending a cue point editing command—an instruction to the computer system to edit a portion of the annotation—because the computer system responds by carrying out the requested action (*see* limitation [13.5] below).

177. Fegan describes the computer system’s receipt of such commands over the computer network (*e.g.*, Internet).

The systems and methods of the present invention may be used to publish stored multimedia presentations by making them available for access by others over a computer network (*e.g.*, **the Internet**). For example, a first user in a first location may create a multimedia presentation that is stored entirely on a **remote**

server. In preferred embodiments of the present invention, one or more additional users in one or more separate and distinct locations on a computer network may access the multimedia presentation created by the first user. In some embodiments, one or more additional users can contribute their own multimedia content, segment information, and annotative information to a multimedia presentation originally created by a first user. Because the systems and methods of the present invention utilize time-based index points and annotations, multimedia presentations may be created, stored, and accessed without the need to permanently store multimedia files. **In addition, the server-side approach of the present invention eliminates the necessity of storing any of the data associated with the multimedia presentation on a user's local computer.**

Id., ¶[0085] (emphasis added).

As used herein, the term **“client-server”** refers to a model of interaction in a distributed system in which a program at one site sends a request to a program **at another site** and waits for a response. The requesting program is called the “client,” and the program that responds to the request is called the “server.” In the context of the World Wide Web (discussed below), the client is a “Web browser” (or simply “browser”) that runs on a computer (e.g., desktop, cell phone, hand-held, etc.) of a user; the program which responds to browser requests by serving Web pages is commonly referred to as a “Web server.”

Id., ¶[0037] (emphasis added).

Further, the client is at the client (*i.e.*, non-server) side of the network and is remote from the server because such intercommunication is over the Internet. *Id.*, ¶[0037].

6. **[13.5] modifying a portion of the cue point information at the server side of the public network according to the cue point editing command received over the public network from the client terminal;**

178. In my opinion, Fegan discloses limitation [13.5].

179. Fegan discloses the server side of the public network modifying the at least a portion of the cue point information (*e.g.*, annotation information) according to the cue point editing command received during limitation [13.4]. In Fegan, when a user at the client makes changes in “Edit” mode and selects “Update,” the edited annotation information is sent over the network (*e.g.*, Internet) to modify the annotation at the server. *See* my discussion above of limitation [13.4]; *see also*:

... **FIG. 7** shows a **saved annotation** that has been opened via the directory for further editing. In preferred embodiments, annotations may be freely created, edited, and deleted without limitation.

Fegan, ¶[0081] (emphasis added); *see also* ¶[0080].

The systems and methods of the present invention may be used to create user-customizable multimedia presentations comprising multimedia files, segment information, and annotative information. In preferred embodiments, **software that is stored on a remote server allows a user to save and store segment and annotative information** associated with one or more

multimedia files. Because the systems and methods of the present invention utilize time and/or spatial-based index points and annotations, multimedia presentations may be created without the need to permanently store multimedia files, or divide them into separate files corresponding to each selected “clip.” In some embodiments, software that is stored on a remote server provides one or more templates upon which a multimedia presentation may be based. In preferred embodiments, both preset and customizable templates are provided. In other embodiments, the multimedia presentation is executed by means of a specialized programming language. In particularly preferred embodiments, the multimedia presentation is executed by means of Synchronized Multimedia Integration Language (“SMIL”). In yet other embodiments, the multimedia presentation is executed by means of Hypertext Markup Language (“HTML”), although any suitable programming language may be used.

Id., ¶[0083]; *see also* ¶¶[0082]-[0085]; FIG. 7.

7. **[13.6] receiving a cue point selection over the network from the remote client terminal identifying one of the cue points corresponding to a portion of the video information of one of the video files; and**

180. In my opinion, Fegan discloses limitation [13.6].

181. Fegan discloses that the client sends, and the remote server receives, a cue point selection (*e.g.*, receiving a selected annotation using a client-generated playlist listing one or more selected annotations) over the network. As I discuss

above, Fegan's annotations correspond to a portion of the video information of one of the video files (e.g., start and stop points within a video or streaming media). See my discussion above of limitation [13.3]. And Fegan explains that the server receives these annotations because it receives playlist selection information including multimedia file identity and multimedia file start and stop points.

Thus, in some embodiments, the present invention provides a system for manipulating, annotating, and managing a plurality of remote, multimedia files in a server-side environment, the system comprising a host computer network configured to carry out one or more of the following: a) identify multimedia files on an Internet web site, b) present the multimedia files to a client computer, c) **receive playlist selection information** (i.e., information pertaining to the identity of the multimedia files to be selected, annotated, or otherwise designated or manipulated) from the client computer, **said playlist selection information comprising multimedia file identity and multimedia file start and stop points (i.e., designations of the starting and ending points of the multimedia file to be played in a presentation)**, d) receive multimedia file annotation information from the client computer, e) catalog (i.e., organize in a database or similar format) playlist selection information and multimedia file annotation information from a plurality of web sites selected by the client computer, and f) combine those multimedia files and annotations into online publications.

Fegan, ¶[0012] (emphasis added).

182. That is, Fegan’s remote server receives a playlist of annotations and uses the annotations to generate a presentation of the video clips the user has selected (*i.e.*, a compilation of each selected multimedia file from its annotated start point to end point). Fegan’s system “uses the url/uri and time offsets to play the portion of the digital object that the user has selected.”

The systems and methods of the present invention avoid these problems by, in preferred embodiments, only storing the url/uri (a web pointer) to the digital object. To isolate a piece of the digital object, the user only has to press “begin recording” and “end recording.” This—again—does not store bits of the digital object, but simply the time offsets that the user has selected. The application then uses the url/uri and time offsets to play the portion of the digital object that the user has selected, and provides a means to add the user’s own thinking to those portions of media. The technical tools for carrying out the multimedia presentation are maintained on the server-side, rather than forcing the user on the client-side to navigate the difficulties of the management of the multimedia files.

Id., ¶[0007] (emphasis added).

... For example, in some embodiments, the system selects an appropriate multimedia player (e.g., with selection based on the nature of the content to be played) and finds and loads the desired media into an editor. The media is displayed to the user along with annotation tools that allow the user to isolate portions of the

media relevant to the presentation and to annotate the media as desired. **In some embodiments, each user has a personal “portal” hosted on the system of the present invention that allows the user to edit and organize collected materials and to create complex publications and presentations that include multimedia clips and annotations.**

Id., ¶[0009] (emphasis added).

183. Since the client computer connects to the remote server via a “computer network (e.g., the Internet),” the cue point selection is received over the network (e.g., Internet). *Id.*, ¶¶[0071], [0008].

184. Each annotation (cue point) identifies a portion of the video, by identifying a video start and stop point.

Thus, in addressing the limitations of the prior art, the present invention provides comprehensive systems for identifying, segmenting, collecting, annotating, and publishing multimedia materials. In preferred embodiments, the systems and methods are hosted in an entirely server-side environment, such that client computers can identify, segment, collect, annotate, organize, and publish complex multimedia presentations containing multimedia files or portions of multimedia files from a plurality of sources (e.g., any web site offering multimedia on the internet). In some such preferred embodiments, **the identity of the files, the annotations, and the organization of the files is stored and managed on the host network.** ...

Id., ¶[0008] (emphasis added).

As shown in FIG. 4, a user-selected multimedia file and compatible multimedia player are directed by software to be displayed in a browser window. In preferred embodiments, the browser window contains additional tools for segmenting multimedia files. In particularly preferred embodiments, the segmenting tools allow a user to generate **time-based index points for a multimedia file** (e.g., spatial parameters for image files). The use of time-based/spatial-based index points provides a means for the virtual segmentation of a multimedia file, allowing a user to create “clips” of specified portions. Playback of the multimedia file may be started or stopped at any point in the file's timeline. **The data associated with the time-based index points may be saved and stored on a remote server, and may be retrieved from the remote server and used whenever the associated multimedia file is accessed.** Because the systems and methods of the present invention utilize time-based/spatial-based index points, multimedia editing may be achieved without the need to permanently store multimedia files, or divide them into separate files corresponding to each selected “clip.” The only data required to be stored for the purpose of segmenting multimedia files is the data associated with the time-based index points. ...

Id., ¶[0078] (emphasis added); *see also* ¶[0010] (server-side databases for “stop and start points”), [0007], [0012].

185. The playlist selection containing annotations identifies the selected cue point.

Thus, in some embodiments, the present invention provides a system for manipulating, annotating, and managing a plurality of remote, multimedia files in a server-side environment, the system comprising a host computer network configured to carry out one or more of the following: a) identify multimedia files on an Internet web site, b) present the multimedia files to a client computer, c) **receive playlist selection information** (i.e., information pertaining to the identity of the multimedia files to be selected, annotated, or otherwise designated or manipulated) from the client computer, said playlist selection information **comprising multimedia file identity and multimedia file start and stop points (i.e., designations of the starting and ending points of the multimedia file to be played in a presentation)**, d) receive multimedia file annotation information from the client computer, e) catalog (i.e., organize in a database or similar format) playlist selection information and multimedia file annotation information from a plurality of web sites selected by the client computer, and f) combine those multimedia files and annotations into online publications.

Id., ¶[0012] (emphasis added).

8. **[13.7] streaming the portion of the video information to the client terminal over the public network in response to receiving the cue point selection.**

186. In my opinion, Fegan discloses limitation [13.7].

187. Fegan discloses that in response to receiving the cue point selection (e.g., a playlist containing one or more annotations), streaming the associated portion of the video information (e.g., streaming, to the client computer, the portion of the video which corresponds to the selected annotation) to the client terminal over the public network (e.g., Internet). This is done in response to receiving the cue point selection (e.g., receiving the selected annotation, selected from client-generated playlist in limitation [13.6]). That is, the remote server creates a presentation from the playlist it receives from the client computer and streams it to the client. *See my discussion above of limitation [13.6]; see also:*

Thus, in some embodiments, the present invention provides a system for manipulating, annotating, and managing a plurality of remote, multimedia files in a server-side environment, the system comprising a host computer network configured to carry out one or more of the following: a) identify multimedia files on an Internet web site, b) present the multimedia files to a client computer, c) **receive playlist selection information** (i.e., information pertaining to the identity of the multimedia files to be selected, annotated, or otherwise designated or manipulated) from the client computer, said playlist selection information **comprising multimedia file identity and multimedia file start and stop points** (i.e., designations of the starting and ending points of the **multimedia file to be played in a presentation**), d) receive multimedia file annotation information from the client computer, e) catalog (i.e., organize in a database or similar

format) playlist selection information and multimedia file annotation information from a plurality of web sites selected by the client computer, and f) combine those multimedia files and annotations into online publications.

Fegan, ¶[0012] (emphasis added).

... The application then **uses the url/uri and time offsets to play the portion of the digital object that the user has selected**, and provides a means to add the user's own thinking to those portions of media. The technical tools for carrying out the multimedia presentation are maintained on the server-side, rather than forcing the user on the client-side to navigate the difficulties of the management of the multimedia files.

Id., ¶[0007] (emphasis added).

The selected multimedia file and compatible multimedia player are then displayed in a browser window. FIG. 4 shows a browser window in which a user-selected multimedia file and a compatible multimedia player are displayed. ...

Id., ¶[0076] (emphasis added).

The systems and methods of the present invention may be used to publish stored multimedia presentations by making them **available for access by others over a computer network (e.g., the Internet)**.

Id., ¶[0085] (emphasis added); *see also* ¶¶[0082]-[0084].

... One advantage of this system is the ability to publish and present multimedia events with no proliferation of the digital files encoding the multimedia events (i.e., embodiments of the present invention provide a pointer to the media rather than a copy of the media).

Id., ¶[0010] (emphasis added).

The present invention is not limited by the nature of the multimedia files. For example, **multimedia files** include, but are not limited to, **streaming media**, audio information, **video information**, image information, and text information.

Id., ¶[0013] (emphasis added); *see also* ¶[0033].

3. The system of claim 1, wherein said multimedia files comprise **streaming media**.

Fegan, claim 3 (emphasis added).

C. Claim 14

188. Claim 14 recites:

14. The method of claim 13, further comprising providing a cue point editing mode as part of the video editing environment configured to perform at least one of creating, modifying, deleting, or interpreting at least a portion of the set of cue point information according to the cue point editing command received over the public network from the client terminal.

189. In my opinion, Claim 14 is anticipated by Fegan.

190. Fegan discloses providing a cue point (annotation) editing mode as part of the video editing environment, by providing interfaces for editing annotations as part of software executed by the remote server and databases, including when a user selects the “Edit” control button for a selected annotation. *See* my discussion above of [13.4], where I explain how selecting “Edit” from the left pane for a selected annotation provides an “edit mode” for the selected annotation (FIG. 7).

191. Fegan further describes that this “Edit” interface (FIG. 7) is configured to at least modify at least a portion of the set of cue point (annotation) information according to the cue point editing command received over the public network (Internet) from the client terminal. For example, FIG. 7 shows a user being provided with options to modify of annotation information such as “start position,” “end position,” “title,” and “annotation”, and FIG. 7 also illustrates an “Update” button (cue point editing command) which, when selected, sends the modified information to the remote server.

... FIG. 6 shows a browser window in which saved annotative information has been selected from the directory shown in FIG. 5. When a user selects an annotation, data comprising the annotative information is displayed in the browser window, allowing the user to determine the contents of an annotation prior to opening it for editing. **FIG. 7 shows a saved annotation that has been opened via the directory for further editing.** In

preferred embodiments, annotations may be freely created, edited, and deleted without limitation.

Fegan, ¶[0081] (emphasis added).

192. Fegan's remote server modifies at least a portion of the annotation information according to the editing command received over the network from the client computer. *See* my discussion above of limitations [13.4]-[13.5]; *see also*:

The systems and methods of the present invention may be used to create user-customizable multimedia presentations comprising multimedia files, segment information, and annotative information. In preferred embodiments, software that is stored on a remote server allows a user to **save and store segment and annotative information** associated with one or more multimedia files. ...

Id., ¶[0083] (emphasis added).

D. Claim 15

193. Claim 15 recites:

15. The method of claim 14, further comprising: receiving an edit mode entry signal over the network from the client terminal representing entry into the cue point edit mode, and switching the video editing environment to the cue point editing mode in response to receiving the edit mode entry signal.

194. In my opinion, Claim 15 is anticipated by Fegan.

195. Fegan discloses receiving an edit mode entry signal (selection of “Edit” in the FIGs. 5-6 interfaces, for entry into the interface(s) for editing the selected annotation) over the network from the client terminal representing entry into the cue point edit mode and switching the video editing environment to the cue point editing mode (FIG. 7 interface) in response to receiving the edit mode entry signal. *See my discussion above of limitation [13.4] and claim 14; see also:*

As used herein, the term “HTTP” refers to HyperText Transport Protocol that is the standard World Wide Web **client-server protocol used for the exchange of information (such as HTML documents, and client requests for such documents) between a browser and a Web server**. HTTP includes a number of different types of messages that can be sent from the client to the server to request different types of server actions. For example, a “GET” message, which has the format GET, causes the server to return the document or file located at the specified URL.

Fegan, ¶[0044] (emphasis added).

196. In particular, a web browser user interface, where server-side information is to be displayed on the interface screen, involves an HTTP request by the client seeking retrieval of such information stored on the server side, whereby in response, a web server provides HTML web page data to present such information on the client web browser. Notably, since Fegan teaches that the annotations are

“stored and managed on the host network” (Fegan, ¶[0008]) rather than being stored at the client side (*id.*, ¶[0085]), an HTTP request corresponding to the edit mode entry signal must be sent by a client (and received by the computer system), so that the information for editing by the user (as seen in the FIG. 7 interface) is retrieved at the server side and presented in the FIG. 7 interface.

197. A POSITA would have understood the plain meaning of “edit mode entry signal” to be a signal, sent from the remote client terminal and received by the computer system, to enter an edit mode. Further, a POSITA would have understood the word “signal” (in “edit mode entry signal”) to mean an indication. *See* Exhibit 1018 (“Signal.” Merriam-Webster.com Dictionary, Merriam-Webster, <https://www.merriam-webster.com/dictionary/signal> (accessed March 29, 2024) (defining “signal” as “1: SIGN, INDICATION”)).

198. As the ’286 Patent explains, “in some embodiments, cue point edit mode will always be available, while other embodiments may require **active entry into that mode, for example by polling for input.**” ’286 Patent, 15:28-31 (emphasis added). The “input” may be “a particular event from an input device, like a mouse or keyboard.” *Id.*, 15:31-33; *see also id.*, 18:42:49 (explaining “the client may enter the cue point edit mode in any useful way” including “click[ing] on a video image” or “select[ing] a GUI control dedicated to providing entry into the cue point edit mode” or “select[ing] a portion of the video progress bar”).

199. Therefore, a POSITA would have understood that when a user selects the “Edit” option in interface of Fig. 5 for a selected annotation, the remote server receives an edit mode entry signal—an indication to the computer system to enter an edit mode.

E. Claim 18

200. Claim 18 recites:

18. The method of claim 13, further comprising receiving, over the network, authentication information relating to access authority of the remote client terminal.

201. In my opinion, Claim 18 is anticipated by Fegan.

202. Fegan discloses receiving, over the network (Internet), authentication information relating to access authority of the remote client terminal. Fegan describes that “each user has a personal ‘portal’ hosted on the system” that “allows the user to edit and organize collected materials to create complex publications and presentations that include multimedia clips and annotations.”

... For example, in some embodiments, the system selects an appropriate multimedia player (e.g., with selection based on the nature of the content to be played) and finds and loads the desired media into an editor. The media is displayed to the user along with annotation tools that allow the user to isolate portions of the media relevant to the presentation and to annotate the media as desired. In some embodiments, **each user has a personal “portal” hosted on the system of the present invention that**

allows the user to edit and organize collected materials and to create complex publications and presentations that include multimedia clips and annotations.

Fegan, ¶[0009] (emphasis added).

After a user designates a location for storage and retrieval of the annotative information, the information appears on the **user's personal portal page** that can be accessed through a browser window. FIG. 5 shows a browser window displaying a user-created directory of annotative information associated with multimedia files. In preferred embodiments of the present invention, the directory allows a user to select, open, edit, close, or delete the annotative information associated with a multimedia file. FIG. 6 shows a browser window in which saved annotative information has been selected from the directory shown in FIG. 5. ...

Id., ¶[0081] (emphasis added).

As used herein, the term “security protocol” refers to an electronic security system (e.g., hardware and/or software) to limit access to processor to specific users authorized to access the processor. For example, a security protocol may comprise a software program that **locks out one or more functions of a processor until an appropriate password is entered.**

Id., ¶[0050] (emphasis added).

203. Providing a user with its own server-side portal in Fegan involves the server receiving over the network from the client some form of authentication information (whether a username/password, IP address, public/private key, biometrics, etc.) to confirm that the user has access to that particular portal page, and thus granting or denying access based upon that authentication information. This type of authentication information could include automatically generated information (*e.g.*, IP address of the requesting computer or a public or private key) or manually entered information (*e.g.*, username/password, biometrics). These methods of user authentication were common practice in networking at that time, and a POSITA would have understood that Fegan is referring to such user authentication when it references terminology such as a web-based “personal portal” page.

F. Claim 19

204. Claim 19 recites:

19. The method of claim 18, further comprising denying or allowing access to all or part of the video editing environment based at least in part on the access authority of the remote client terminal.

205. In my opinion, Claim 19 is anticipated by Fegan.

206. Fegan discloses denying or allowing access to all or part of the video editing environment based at least in part on the access authority of the remote client

terminal. Fegan describes denying access to part of the software executed by servers 10-13 and stores 17-18, as users are denied full access to a “portal” that is not their own. Fegan describes that “each user has a **personal** ‘portal’ hosted on the system.” That is, the portal is for that user only and other users do not have the same access. See my discussion above of claim 18; *see also*:

... For example, in some embodiments, the system selects an appropriate multimedia player (e.g., with selection based on the nature of the content to be played) and finds and loads the desired media into an editor. The media is displayed to the user along with annotation tools that allow the user to isolate portions of the media relevant to the presentation and to annotate the media as desired. In some embodiments, **each user has a personal “portal” hosted on the system of the present invention that allows the user to edit and organize collected materials and to create complex publications and presentations that include multimedia clips and annotations.**

Fegan, ¶[0009] (emphasis added).

After a user designates a location for storage and retrieval of the annotative information, the information appears on the user’s personal portal page that can be accessed through a browser window. FIG. 5 shows a browser window displaying a user-created directory of annotative information associated with multimedia files. In preferred embodiments of the present invention, the directory allows a user to select, open, edit, close,

or delete the annotative information associated with a multimedia file. FIG. 6 shows a browser window in which saved annotative information has been selected from the directory shown in FIG. 5. ...

Id., ¶[0081] (emphasis added).

G. Claim 20

207. Claim 20 recites:

The method of claim 13, wherein the server video data is streamed progressively to a video playback environment at the remote client terminal, allowing the video player to begin displaying streamed server video data while remaining server video data continues to be streamed.

208. In my opinion, Claim 20 is anticipated by Fegan.

209. Fegan discloses that the server video data is streamed progressively to a video playback environment at the remote client terminal, allowing the video player to begin displaying streamed server video data while remaining server video data continues to be streamed (*i.e.* renders as received). *See* my discussion above of limitation [13.7]; *see also*:

The present invention is not limited by the nature of the multimedia files. For example, **multimedia files include**, but are not limited to, **streaming media**, audio information, video information, image information, and text information.

Fegan, ¶[0013] (emphasis added).

As used herein, the term “client-server” refers to a model of interaction in a distributed system in which a program at one site sends a request to a program at another site and waits for a response. The requesting program is called the “client,” and the program that responds to the request is called the “server.” In the context of the World Wide Web (discussed below), the **client is a "Web browser" (or simply "browser") that runs on a "computer"** (e.g., desktop, cell phone, hand-held, etc.) of a user; the program which responds to browser requests by serving Web pages is commonly referred to as a “Web server.”

Id., ¶[0033] (emphasis added); *see also* ¶¶[0007], [0033], [0057].

A POSITA would have understood the word “progressively” (in “streamed progressively”) to mean streaming that allows display of the streamed server video data to begin while the remaining server video data is still being streamed, based on both the claim language and the ’286 Patent. *See* claim 20 (“streaming progressively ... allowing the video player to begin displaying streamed server video data while remaining server video data continues to be streamed.”); ’286 Patent 25:41-47 (“Further, video files may load in various ways, including as a complete file, streaming, or progressive. Progressive loading of video files, for example, allows the video frame to begin playing the portion of video data which has already loaded while the remainder of video data continues to load.”), 4:7-10 (“client video data is

transmitted progressively, allowing the video player to begin displaying transmitted client video data while remaining client video data continues to be transmitted.”).

XII. GROUND 3: CLAIMS 13-15 AND 18-20 ARE OBVIOUS OVER KIM

A. Summary of Kim

210. Kim (Exhibit 1006), is a publication of a U.S. patent application, published on September 22, 2005. I understand that Kim constitutes prior art at least under pre-AIA 35 U.S.C. §§ 102(a) and (e) even under the alleged May 7, 2006 priority date.

211. Kim describes a client-server architecture comprising “[a] multimedia bookmark (Vmark) bulletin board service (BBS) system” and describes methods and systems for making and sharing multimedia bookmarks using a BBS server, web host, and media host. Kim, Abstract. Kim discloses that an administrator, which can also be a client, can create and edit multimedia bookmarks. *Id.*, ¶¶[0334]-[0336]. Kim explains that the multimedia bookmarks are used to select where to begin streaming the video, for example, “[b]y selecting the play control button 612, the user can watch the video from the bookmarked position.” *Id.*, ¶¶[0295], [0112].

B. Claim 13

212. Claim 13 recites:

[13.P] 13. A method for handling video cue points, the method comprising:

[13.1] providing a computer system having at least one computer-readable medium comprising a video editing environment, and being disposed at a server side of a public network;

[13.2] receiving a set of video information at the video editing environment using the computer system;

[13.3] receiving a set of cue point information at the video editing environment using the computer system;

[13.4] receiving a cue point editing command over the network from a remote client terminal disposed at a client side of the public network using the computer system;

[13.5] modifying a portion of the cue point information at the server side of the public network according to the cue point editing command received over the public network from the client terminal;

[13.6] receiving a cue point selection over the network from the remote client terminal identifying one of the cue points corresponding to a portion of the video information of one of the video files; and

[13.7] streaming the portion of the video information to the client terminal over the public network in response to receiving the cue point selection.

213. In my opinion, Claim 13 is obvious over Kim.

1. [13.P] A method for handling video cue points, the method comprising:

214. In my opinion, to the extent the preamble is limiting, Kim discloses limitation [13.P].

215. Kim teaches a method for handling (“creating ... displaying ... selecting”) a “multimedia bookmark that is composed of ... a video URI, a start time, duration”) video cue points (e.g., “multimedia bookmark” or “Vmark”).

A multimedia bookmark (Vmark) bulletin board service (BBS) system comprises: a web host comprising storage for messages, a web server, and a Vmark BBS server; a media host comprising storage for audiovisual (AV) files, and a streaming server; a client comprising storage for Vmark, a web browser, a media player and a Vmark client; and a Vmark server located at the media host or at the client; a communication network connecting the web host, the media host and the client. A method of performing a multimedia bookmark bulletin board service (BBS) comprises: **creating** a message including a multimedia bookmark for an AV file; and posting the message into the multimedia bookmark BBS. A method of sending multimedia bookmark (Vmark) between clients comprises: at a first client, making a Vmark indicative of a bookmarked position in an AV program; sending the Vmark from the first client to a second client; and playing the program at the second client from the bookmarked position. A system for sharing multimedia content

comprises: a multimedia bookmark bulletin board system (BBS);
and means for posting a multimedia bookmark to the BBS.

Kim, Abstract (emphasis added).

The present disclosure relates to multimedia bookmark and an electronic bulletin board system (hereinafter referred to as a “BBS”) on computer networks. As used in this disclosure, the term **multimedia bookmark** includes **video bookmark (Vmark)**.

Id., ¶[0012] (emphasis added).

More specifically for the posting method, a process for **creating** a message including **multimedia bookmark (herein referred to as a “multimedia bookmark message”)** is provided. Herein, the process includes sub-processes for **displaying** the multimedia bookmark stored in the storage, **selecting** one of the multimedia bookmarks to be posted, and creating the message including multimedia bookmark that is composed of an image data (hereinafter referred to as a “Multimedia Bookmark Image”), a video URI, a start time, duration, and a title page URI of the video.

Id., ¶[0207] (emphasis added); *see also id.*, Claims 6, 12.

216. Kim describes a client-server architecture with a web host comprising storage for messages, a web server, a Vmark (bulletin board service) BBS server, a media host comprising storage for audiovisual (AV) files, a streaming server, a

Vmark server located at the media host, and a communication network connecting the web host, the media host and the client.

A multimedia bookmark (VMark) bulletin board service (BBS) system comprises: a **web host comprising storage** for messages, a **web server**, and a **Vmark BBS server**; a **media host** comprising storage for audiovisual (AV) files, and a **streaming server**; a client comprising storage for Vmark, a web browser, a media player and a Vmark client; and a **Vmark server** located at the media host or at the client; a **communication network connecting the web host, the media host and the client**. A method of performing a multimedia bookmark bulletin board service (BBS) comprises: creating a message including a multimedia bookmark for an AV file; and posting the message into the multimedia bookmark BBS. A method of sending multimedia bookmark (Vmark) between clients comprises: at a first client, **making a Vmark indicative of a bookmarked position in an AV program**; sending the Vmark from the first client to a second client; and **playing the program at the second client from the bookmarked position**. A system for sharing multimedia content comprises: a multimedia bookmark bulletin board system (BBS); and means for posting a multimedia bookmark to the BBS.

Id., Abstract (emphasis added).

217. Kim describes that its multimedia bookmark stores the relative time or byte position from the beginning of an AV file along with the file name, and URI,

and is used to allow a user to access the content at a later time from the position of the multimedia file that the user or any other people have specified.

Multimedia bookmark for AV content is a functionality that allows a user to access the content at a later time from the position of the multimedia file that the user or any other people have specified. The multimedia bookmark stores the relative time or byte position from the beginning of an AV file along with the file name, and URI. Additionally the multimedia bookmark can also store an image extracted from the multimedia bookmark position marked by a user such that the user can easily reach the segment of interest through the title of the multimedia bookmark displayed along with the stored image of the corresponding location. Also, the multimedia bookmark for an AV content which is marked by a user can be transferred to other people by an electronic mail, thus any people receiving the e-mail can play the video from the exact point marked by the user.

Id. ¶[0112] (emphasis added).

218. Kim's multimedia bookmarks are cue points, which the '286 Patent describes as "indexed time positions or tags anywhere within a video file" and may include "meta-tags" that "describe the information in the video file." '286 Patent, 7:50-52, 8:15-20.

219. Therefore, Kim's bookmarks are "cue points" in the context of the '286 Patent claims.

2. **[13.1] providing a computer system having at least one computer-readable medium comprising a video editing environment, and being disposed at a server side of a public network;**

220. In my opinion, Kim discloses limitation [13.1].

221. Kim teaches providing a computer system with servers and storage devices at a server side of a public network, and including a web host 210 and media host 220, along with their servers and storages.

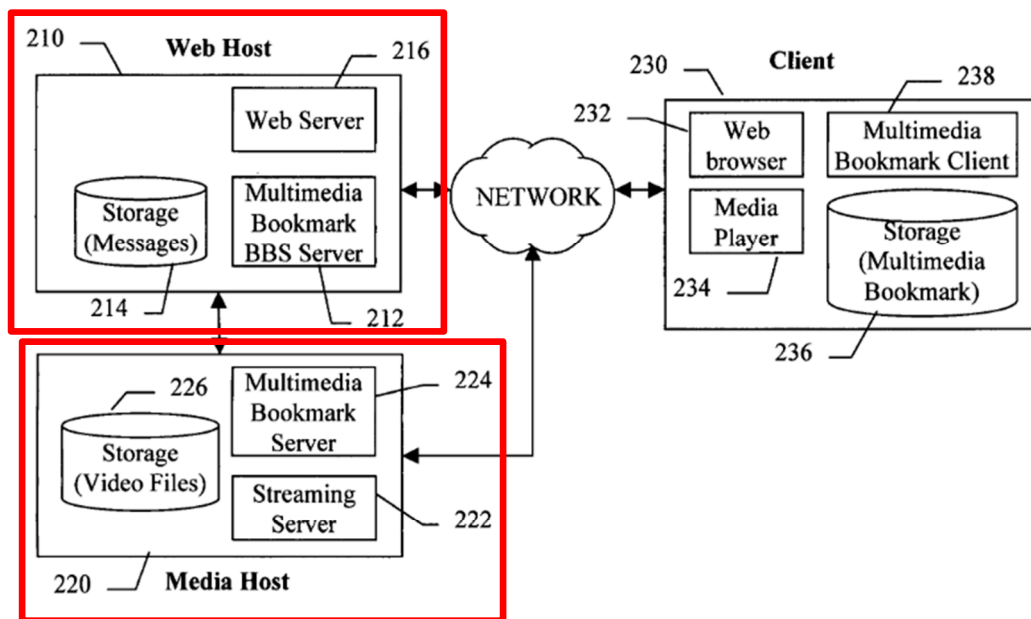


FIG. 2

Id., FIG. 2 (annotated).

Generally, according to the disclosure, the multimedia bookmark BBS comprises a multimedia bookmark BBS server, a multimedia bookmark server, and multimedia bookmark clients located in web host, media host and client computers respectively.

Id., ¶[0206]; *see also* Abstract.

222. Each server/storage has at least one computer-readable medium (*e.g.*, server memory/storage) for executing software to enable the requisite hardware/software modules of the servers to run. For example, Kim describes the multimedia bookmark server 224 as “running at media host 220 in FIG. 2.

From the view point of location where a multimedia bookmark image is captured from a video, there are two ways to capture the bookmark image: one is to utilize **multimedia bookmark server 224 running at media host 220** in FIG. 2 to capture a multimedia bookmark image from the video stored at storage 226 and send the captured bookmark image to requesting client 230, and the other is to utilize multimedia bookmark client 238 running at client computer 230 to capture a multimedia bookmark image directly from a frame buffer of media player 234 playing the video.

Id., ¶[0326] (emphasis added); *see also id.*, claim 9, FIG. 1.

223. Kim describes that these computer-readable media comprise a video editing environment, including software for both editing bookmarks and streaming video, executed by web host 210 and media host 220, and their servers and storages.

224. For example, Kim's video editing environment (which includes the functionality within multimedia bookmark BBS server 212) executes software for editing bookmarks.

The **multimedia bookmark image** (which is a reduced frame of the video captured at the bookmarked position) **might be obtained by bookmark server 224 of media host 220 and then delivered to multimedia bookmark client 238 of client 230.** Multimedia bookmark BBS server 212 communicates with bookmark server 224 in response to a user's request of capturing a multimedia bookmark image. After receiving the captured multimedia bookmark image from the server 224, **bookmark BBS server 212 sends the multimedia bookmark image together with multimedia bookmark information to the web browser 232 of client 230.**

Id., ¶[0286] (emphasis added).

The **multimedia bookmark** saved at local storage 236 of client 230 regardless of whether its multimedia bookmark image is obtained by the bookmark server 224 of media host 220 or bookmark client 238 of client 230 **can be uploaded to multimedia bookmark BBS server 212 of web host 210 and then stored at storage 214 of the web host** so that other users

can share the multimedia bookmark. Thus, one who retrieves the bookmark from multimedia bookmark BBS server 212 can start to play the video exactly from the bookmarked position.

Id., ¶[0288] (emphasis added).

FIG. 18B illustrates an exemplary GUI screen to list the advertising multimedia bookmarks selected by the administrator. After selecting the advertising multimedia bookmarks, the administrator can verify his/her selection by viewing a list of selected multimedia bookmark image 1832 and its multimedia bookmark information described in information fields 1836 such as video title, file location/name, start-time, duration, and the related Web page URI. **The administrator can edit video title or allowable playing duration in information field 1836.** ...

Id., ¶[0335] (emphasis added).

225. Kim's video editing environment (which includes the functionality within multimedia bookmark server 224) also performs a function of obtaining multimedia bookmark images from the videos and delivering them to a client.

The **multimedia bookmark image** (which is a reduced frame of the video captured at the bookmarked position) **might be obtained by bookmark server 224 of media host 220** and then **delivered to multimedia bookmark client 238 of client 230.** Multimedia bookmark **BBS server 212 communicates with bookmark server 224** in response to a user's request of capturing a multimedia bookmark image. After receiving the

captured multimedia bookmark image from the server 224, bookmark BBS server 212 sends the multimedia bookmark image together with multimedia bookmark information to the web browser 232 of client 230.

Id., ¶[0286] (emphasis added).

When a user of client 2026 makes a multimedia bookmark while playing a video with media player 2028, the client **requests remote multimedia bookmark server 2024 to capture a multimedia bookmark image from the video with information on the user.** Then, before capturing the multimedia bookmark image from the video, multimedia bookmark server 2024 negotiates with license server 2016 and Web server 2018, and requests license server 2016 to retrieve a license key of the user from database 2014. The license server will then return the license key of the user to the multimedia bookmark server if it exists. The multimedia bookmark server then unpacks the requested DRM packaged video stored at storage 2022 with the returned license key of the user, and captures a multimedia bookmark image from the video at a requested bookmarked position. The extracted multimedia bookmark image is sent back to client 2026.

Id., ¶[0345] (emphasis added).

226. Kim's video editing environment (which also includes functionality within streaming server 222) also executes software for streaming video data to the client.

Web host 210 provides lists of videos which have corresponding links or URIs of the videos stored at media host 220 with a hypertext markup language (HTML). **Media host 220 stores the videos in its local storage 226, and provides them to client 230 when they are requested.** Client 230 **can select** one of the videos from the lists displayed in web browser 232. The selection process requests the video to be serviced to media host 220, and **client 230 receives the video streamed by streaming server 222 of media host 220** and displays the video on media player 234.

Id., ¶[0284] (emphasis added).

Media host 220 comprises streaming server 222, bookmark server 224 and storage 226 for archiving media files. Bookmark server 224 is responsible for handling the request from bookmark BBS server 212. The multimedia bookmark server obtains a bookmark image at the required position in accordance with the request and then sends the captured bookmark image to multimedia bookmark BBS server 212 as a reply for the request of multimedia bookmark image. **Streaming server 222 is responsible for the request from client 230 to play a video.**

Id., ¶[0289] (emphasis added).

227. Notably, the '286 Patent broadly describes that a “video editing environment” (or “VEE”) “may comprise any graphical, textual, audio, or other data which is desirable or necessary **for providing interface capabilities to a client**” and “does not [even] require any graphical interface elements” (e.g., even encompasses “script”-level interfaces). '286 Patent, 13:48-67.

228. Kim describes such computer system being disposed at a server-side of a public network (such the Internet), because Kim describes that the web host 210, media host 220 and client 230 are connected by a conventional communication network

FIG. 2 illustrates the general system architecture of the multimedia bookmark BBS, according to an embodiment of the present disclosure. The system comprises multimedia bookmark BBS server 212 located in web host 210, multimedia bookmark server 224 located in media host 220 and multimedia bookmark client 238 located in client 230. **The web host 210, media host 220 and client 230 are connected by a conventional communication network (“NETWORK”).**

Id., ¶[0283] (emphasis added).

Audiovisual (AV) contents are increasingly populated in the **Internet** and there might be many people who want to talk about and share their AV files or AV segments of interest with others. Bulletin board systems enable users to share their messages with others through computer network. ...

Id., ¶[0111] (emphasis added).

The multimedia bookmark for an AV file comprises the following bookmark information:

1. URI of a bookmarked file;
2. Bookmarked position;
3. Content information such as an image captured at a bookmarked position;
4. Textual annotations attached to a segment that contains the bookmarked position;
5. Title of the bookmark;
6. Metadata identification (ID) of the bookmarked file;
7. URI of an opener **web page** from which the bookmarked file started to play;
8. Bookmarked date.

Id., ¶[0264]-[0272] (emphasis added).

FIG. 7 is an exemplary flowchart illustrating the overall method of creating a multimedia bookmark message, posting the message to a multimedia bookmark BBS, and reading the message from the BBS, according to an embodiment of the present disclosure. ... The multimedia bookmark client, which is usually embedded in an **Internet web browser**, reads the list of messages for a message group from multimedia bookmark BBS server 212 of FIG. 2 at step 704, and displays message list

window 300 with additional multimedia bookmark images 310 as shown in FIG. 3. ...

Id., ¶[0297] (emphasis added).

Commonly-owned, copending U.S. patent application Ser. No. 09/911,293 filed Jul. 23, 2001 (Publication No. 2002/0069218) discloses system and method for transferring the multimedia bookmarks between users using e-mails and short message services (SMS). The prior art assumes an environment where videos or video streams are archived at separate sites **connected to the Internet such as media host 220 of FIG. 2**. Bookmark information of a multimedia bookmark includes the URI of a bookmarked video file, which specifies the location of the file that is stored at the sites. Thus, anyone who receives a multimedia bookmark e-mail including the bookmark information can access the video file.

Id., ¶[0349] (emphasis added).

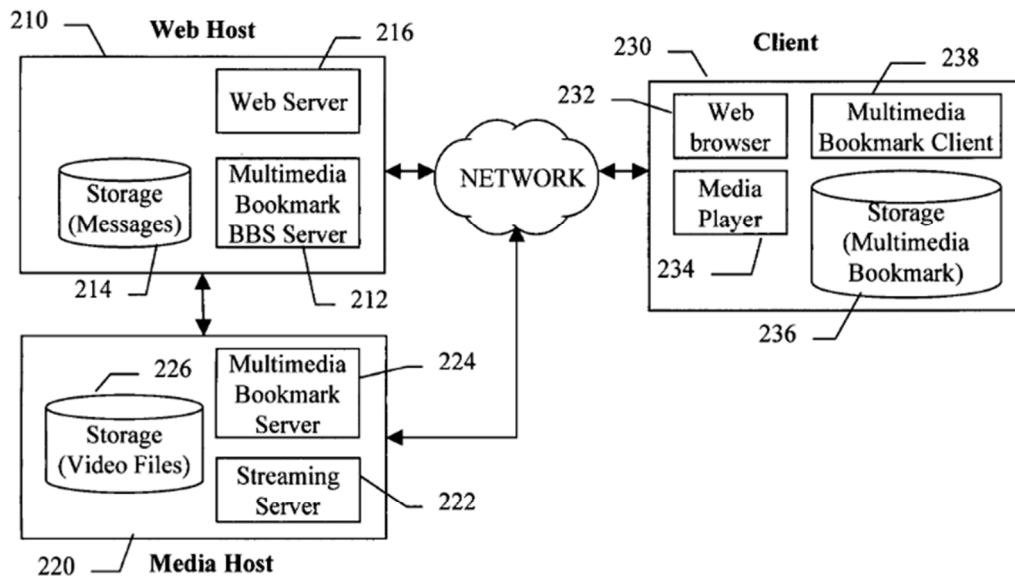


FIG. 2

FIG. 2 (showing a web host 210, web server 216, and web browser 230).

3. [13.2] receiving a set of video information at the video editing environment using the computer system;

229. In my opinion, Kim discloses limitation [13.2].

230. Kim discloses multiple instances of receiving a set of video information (e.g., video media or audiovisual files) at the video editing environment using the computer system (e.g., at the software for streaming video using web host 210, media host 220, and their servers and storages).

231. A first instance of Kim's computer system receiving a set of video information is where the streaming server 222 accesses videos from storage 226 to stream such video data to the client.

Web host 210 provides lists of videos which have corresponding links or URIs of the videos stored at media host 220 with a hypertext markup language (HTML). **Media host 220 stores the videos in its local storage 226, and provides them to client 230 when they are requested.** Client 230 can select one of the videos from the lists displayed in web browser 232. The selection process requests the video to be serviced to media host 220, and client 230 receives the **video streamed by streaming server 222 of media host 220** and displays the video on media player 234.

Id. ¶[0284] (emphasis added).

Media host 220 comprises streaming server 222, bookmark server 224 and storage 226 for archiving media files. Bookmark server 224 is responsible for handling the request from bookmark BBS server 212. The multimedia bookmark server obtains a bookmark image at the required position in accordance with the request and then sends the captured bookmark image to multimedia bookmark BBS server 212 as a reply for the request of multimedia bookmark image. Streaming server 222 is responsible for the request from client 230 to play a video.

Id., ¶[0289] (emphasis added).

According to the techniques disclosed herein, a multimedia bookmark (Vmark) bulletin board service (BBS) system comprises: a web host comprising storage for messages, a web server, and a Vmark BBS server; **a media host comprising storage for audiovisual (AV) files, and a streaming server**; a client comprising storage for Vmark, a web browser, a media player and a Vmark client; and a Vmark server located at the media host or at the client; a communication network connecting the web host, the media host and the client.

Id., ¶[0211] (emphasis added),

Web host 210 provides lists of videos which have corresponding links or URIs of the videos stored at media host 220 with a hypertext markup language (HTML). **Media host 220 stores the videos in its local storage 226, and provides them to client 230 when they are requested.** ... The selection process requests the video to be serviced to media host 220, and client 230 receives the video streamed by streaming server 222 of media host 220 and displays the video on media player 234.

Id., ¶[0284] (emphasis added).

232. A second instance of Kim's computer system receiving a set of video information is where the streaming server 2020 and multimedia bookmark server 2024 access packaged videos.

FIG. 20A illustrates the general system architecture for making multimedia bookmarks on the DRM packaged videos, wherein

multimedia bookmark server module 2024 is running at a remote host computer. In the figure, video encoder 2010 encodes and packages the video source 2012 with DRM. **The DRM packaged video is stored at storage 2022 where the packaged videos are accessed by streaming server 2020 and multimedia bookmark server 2024.** A license key used to unpack the packaged video is stored at database 2014 of license server 2016.

...

Kim, ¶[0344] (emphasis added).

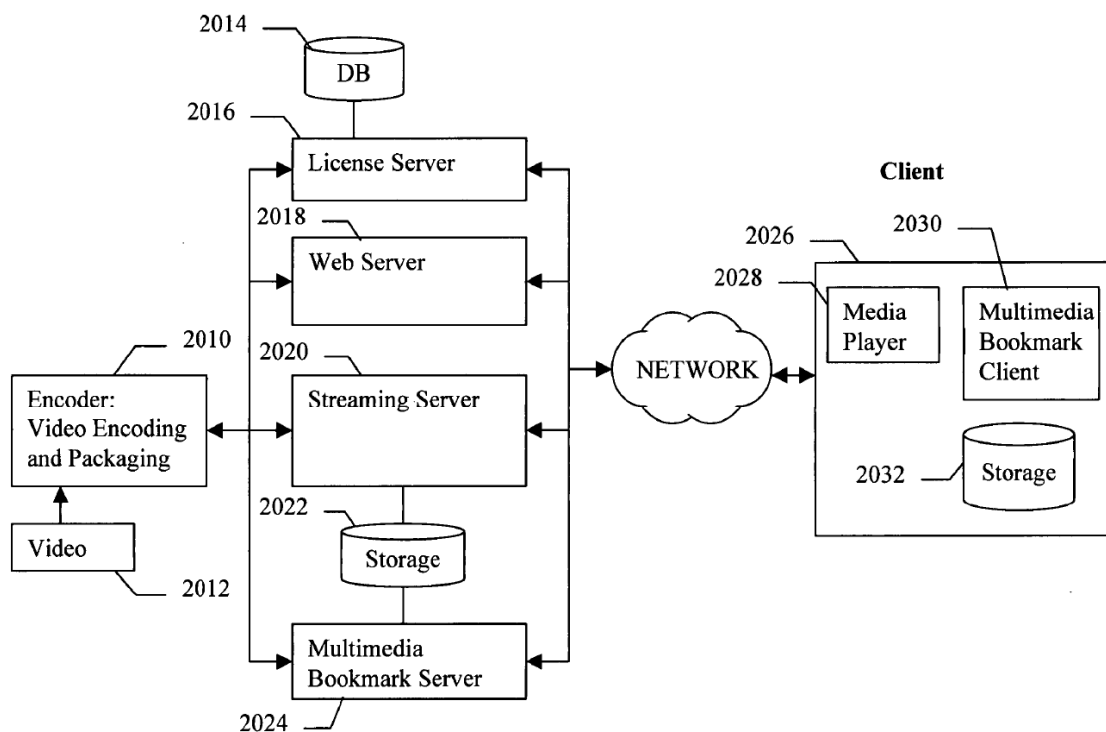


FIG. 20A

FIG. 20A (showing streaming server 2020 connected to storage 2022).

233. Such video information is received at the video editing environment using the computer system because videos are received via the interfaces and program modules of streaming server 222/2020, which are servers in Kim's system used to stream video.

Web host 210 provides lists of videos which have corresponding links or URIs of the videos stored at media host 220 with a hypertext markup language (HTML). Media host 220 stores the videos in its local storage 226, and provides them to client 230 when they are requested. Client 230 can select one of the videos from the lists displayed in web browser 232. The selection process requests the video to be serviced to media host 220, and client 230 receives the **video streamed by streaming server 222 of media host 220** and displays the video on media player 234.

Id., ¶[0284] (emphasis added)

FIG. 1 shows an exemplary GUI screen incorporating the multimedia bookmark of the previous art, that is, commonly-owned, copending U.S. patent application Ser. No. 09/911,293 filed Jul. 23, 2001 (Publication No. 2002/0069218), and additional features, according to an embodiment of the techniques of the present disclosure. **The user interface of the player window 102 is composed of a playback area 112 and a bookmark list 116. Further, the playback area 112 includes a multimedia player 104. ...**

Id., ¶[0275] (emphasis added); *see also* FIG. 1.

4. [13.3] receiving a set of cue point information at the video editing environment using the computer system;

234. In my opinion, Kim discloses limitation [13.3].

235. Kim teaches multiple instances where the computer system (*e.g.*, web host 210 and media host 220, along with their servers and storages) receives a set of cue point information (*e.g.*, a collection of information concerning multimedia bookmarks) at the video editing environment in multiple ways. Notably, the multimedia bookmark “stores the relative time or byte position from the beginning of an AV file along with the file name, and URI.” Kim, ¶[0012].

236. A first instance of Kim’s computer system receiving a set of cue point information at the video editing environment is where the multimedia bookmark BBS server 212 receives a multimedia bookmark that had been created locally at client 230 and uploaded to the multimedia bookmark BBS server.

The multimedia bookmark saved at local storage 236 of client 230 regardless of whether its multimedia bookmark image is obtained by the bookmark server 224 of media host 220 or **bookmark client 238 of client 230 can be uploaded to multimedia bookmark BBS server 212 of web host 210 and then stored at storage 214 of the web host** so that other users can share the multimedia bookmark. Thus, one who retrieves the bookmark from multimedia bookmark BBS server 212 can start to play the video exactly from the bookmarked position.

Id., ¶[0288] (emphasis added).

Generally, according to the disclosure, techniques are provided for posting and retrieving a multimedia bookmark conveniently to and from the multimedia bookmark BBS similar to posting and retrieving a message to and from the conventional BBSs.

Id., ¶[0205] (emphasis added).

Generally, according to the disclosure, a process for storing the transferred multimedia bookmark into the storage of the multimedia bookmark BBS server is also provided.

Id., ¶[0208] (emphasis added).

237. A second instance of Kim's computer system receiving a set of cue point information at the video editing environment is where the multimedia bookmark BBS server 212 receives a multimedia bookmark from storage to allow a client 230 to view the bookmark.

FIGS. 3, 4, 5, and 6 illustrate exemplary GUI screens of a multimedia bookmark BBS, according to an embodiment of the present disclosure. FIG. 3 illustrates an exemplary GUI screen of a message list window 300 of a multimedia bookmark BBS, according to an embodiment of the present disclosure. In the figure, message list window 300 of the multimedia bookmark BBS comprises general components of conventional BBS such as title of a message 312, uploading date of the message 314, and writer of the message 316. Furthermore, message list window 300 of the present disclosure includes multimedia bookmark

of the message 310 for each message. By viewing the visual information of multimedia bookmarks, the multimedia bookmark BBS users can easily identify a message in which they are interested. ...

Id., ¶[0290] (emphasis added).

The multimedia bookmark saved at local storage 236 of client 230 regardless of whether its multimedia bookmark image is obtained by the bookmark server 224 of media host 220 or bookmark client 238 of client 230 can be uploaded to multimedia bookmark BBS server 212 of web host 210 and then stored at storage 214 of the web host so that other users can share the multimedia bookmark. Thus, one who retrieves the bookmark from multimedia bookmark BBS server 212 can start to play the video exactly from the bookmarked position.

Id., ¶[0288] (emphasis added).

If the user wants to post a message at step 712, the "create message" subprocess 714 is to be started with posting window 400 of FIG. 4, and then the "post message" subprocess 716 is also to be started where the multimedia bookmark BBS server receives the message and stores it into the database of storage 214. The detailed sub-processes of both "create message" and "post message" are described with reference to FIGS. 8 and 9, respectively.

Id., ¶[0299] (emphasis added); *see also* FIG. 3.

238. Such cue point information is received at the video editing environment because multimedia bookmark information is received via the interfaces and program modules of the multimedia bookmark BBS server 212 (one of the servers in Kim's system that is used to edit annotations).

Generally, according to the disclosure, a process for **storing** the transferred multimedia bookmark into the **storage of the multimedia bookmark BBS server** is also provided.

Id., ¶[0208] (emphasis added).

The multimedia bookmark saved at local storage 236 of client 230 regardless of whether its multimedia bookmark image is obtained by the bookmark server 224 of media host 220 or bookmark client 238 of client 230 **can be uploaded to multimedia bookmark BBS server 212 of web host 210 and then stored at storage 214 of the web host so that other users can share the multimedia bookmark**. Thus, one who retrieves the bookmark from multimedia bookmark BBS server 212 can start to play the video exactly from the bookmarked position.

Id., ¶[0288] (emphasis added).

If the user wants to post a message at step 712, the "create message" subprocess 714 is to be started with posting window 400 of FIG. 4, and then the "post message" subprocess 716 is also to be started where the **multimedia bookmark BBS server receives the message and stores it into the database of storage 214**. The detailed sub-processes of both "create message" and

"post message" are described with reference to FIGS. 8 and 9, respectively.

Id., ¶[0299] (emphasis added).

5. [13.4] receiving a cue point editing command over the network from a remote client terminal disposed at a client side of the public network using the computer system;

239. In my opinion, Kim discloses or at least renders obvious limitation [13.4].

240. Kim teaches the computer system (*e.g.*, web host 210 and media host 220, along with their servers and storages) receiving a cue point editing command (*e.g.*, an instruction to modify a multimedia bookmark) from a remote client terminal (*e.g.*, client 230) in multiple ways.

241. A first instance of Kim's computer system receiving a cue point editing command over the network from a remote client terminal is where the multimedia bookmark BBS server 212 receives the cue point editing command in response to an administrator selecting a save button in a user interface. For example, FIGs. 18A-18C show exemplary GUI screens displayed on a remote administrator terminal for editing bookmarks. Kim describes that to edit an existing bookmark, an administrator selects the advertising multimedia bookmarks from his/her own multimedia bookmarks and then clicking on save button 1822, using the interface illustrated in FIG. 18A.

To prepare the advertising multimedia bookmarks, the administrator makes his/her own multimedia bookmarks from new or existing videos that he/she wants to advertise, and then selects some multimedia bookmarks to be attached from his/her own multimedia bookmarks. FIGS. 18A, 18B and 18C illustrate exemplary GUI screens of a managing tool for the **administrator to select the advertising multimedia bookmarks from his/her own multimedia bookmarks**, according to an embodiment of the present disclosure. FIG. 18A illustrates an exemplary GUI screen to select advertising multimedia bookmarks from the administrator's own multimedia bookmarks. By typing ID on the ID input text field 1812 and clicking on submit button 1814 in the initial GUI screen 1810, the administrator can view his/her multimedia bookmarks 1818 on multimedia bookmark list box 1816. By checking the check box 1820 below an interesting multimedia bookmark 1818, the administrator selects one advertising multimedia bookmark. By repeating the process, the administrator can select advertising multimedia bookmarks as he/she wants. The administrator then completes the selection by **clicking on save button 1822**, and a new GUI screen 1830 of FIG. 18B will appear.

Kim, ¶[0334] (emphasis added).

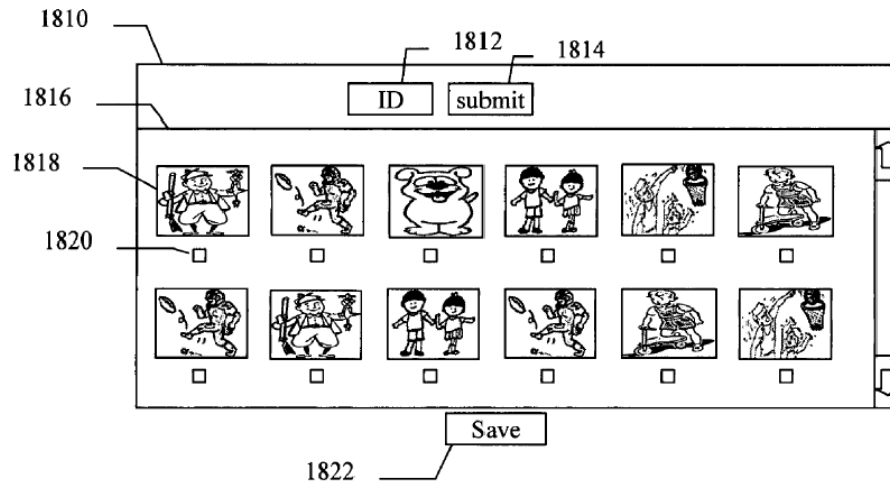


FIG. 18A

Id., FIG. 18A.

242. The administrator can then, using the interface of FIG. 18B, view the selected bookmarks' information described in information fields 1836, such as video title, file location/name, start-time, duration, etc., and can "edit", for example, the video title or allowable playing duration in information field 1836.

FIG. 18B illustrates an exemplary GUI screen to list the advertising multimedia bookmarks selected by the administrator. After selecting the advertising multimedia bookmarks, the administrator can verify his/her selection by viewing a list of selected multimedia bookmark image 1832 and its multimedia bookmark information described in information fields 1836 such as video title, file location/name, start-time, duration, and the related Web page URI. The **administrator** can **edit video title**

or allowable playing duration in information field 1836. Also, the administrator can remove the multimedia bookmark 1832 from the selected list by clicking on delete button 1834. The administrator then completes the verification by clicking on save button 1838, and a new GUI screen 1840 of FIG. 18C will appear.

Id., ¶[0335] (emphasis added); *see also* ¶[0334].

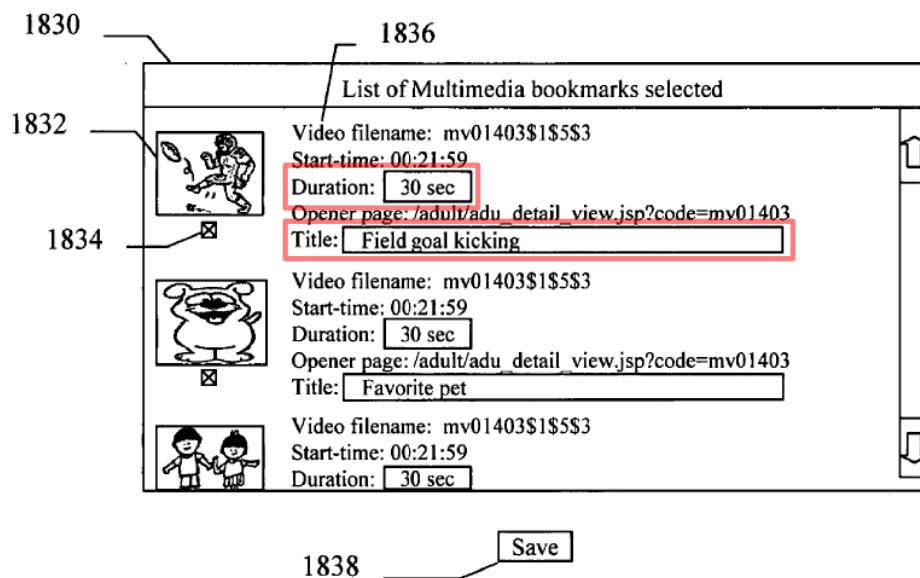


FIG. 18B

Id., FIG. 18B (annotated).

243. Thereafter, the administrator can click on a save button 1838 (illustrated in FIG. 18B, which is reproduced above), verify the edits, and click on a save button 1850, resulting in a cue point editing command being sent to the multimedia

bookmark BBS server 212. As such, the cue point editing command is received using the computer system.

After verifying the advertising multimedia bookmarks, the manager can preview the selected advertising multimedia bookmarks. FIG. 18C illustrates an exemplary GUI screen to preview the advertising multimedia bookmarks selected by the administrator. The preview window 1840 is similar to the advertising multimedia bookmark part 1718 of the multimedia bookmark e-mail 1710 in FIG. 17. The administrator can verify the final format of advertising multimedia bookmarks to be attached to a multimedia bookmark e-mail sent by a user. ... And then, **by clicking the save button 1850** with "Advertising Multimedia Bookmark Mail" radio button 1846 checked, the **advertising multimedia bookmarks are stored on database** to be automatically attached into user's multimedia bookmark e-mail whenever a user sends a multimedia bookmark e-mail to other. **By clicking the save button 1850** with "Advertising News Letter" radio button 1848 checked, the **advertising multimedia bookmarks are stored on database** to be automatically attached into a news letter e-mail whenever a promotion is done by sending a news letter e-mail to users.

Id., ¶[0336] (emphasis added).

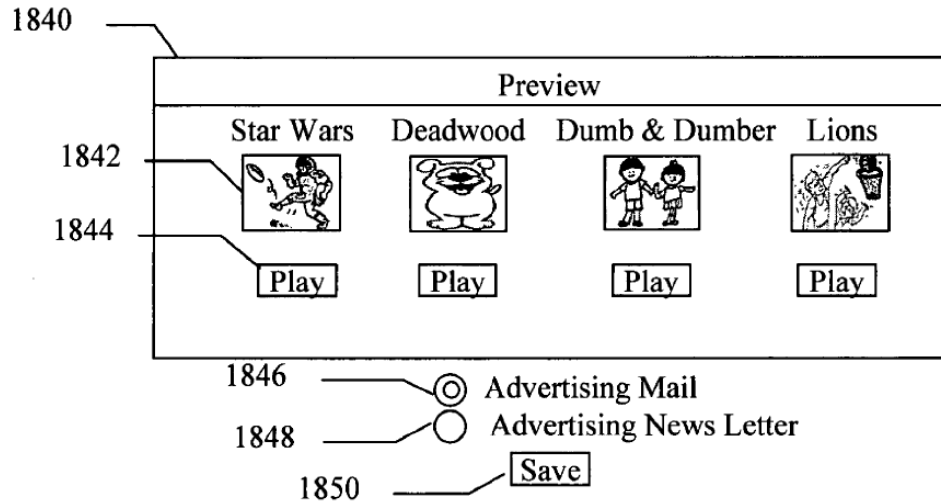


FIG. 18C

Id., FIG. 18C.

FIG. 10 illustrates an exemplary structure of the **multimedia bookmark message** which is **posted to multimedia bookmark BBS server 212 by client 230** of FIG. 2, according to an embodiment of the present disclosure. In the figure, the multimedia bookmark message 1004 in the client 1002 has a body section 1006 and multimedia bookmark section 1008. The body section 1006 includes, what are usually included in the typical BBS, board name, user identifier (user id), the title of the message, and the user description for the message, whereas the multimedia bookmark section 1008 includes multimedia bookmark information such as video URI, title page URI, start time, duration, and multimedia bookmark image data 1010. The multimedia bookmark information is retrieved from the stored multimedia bookmark files 1012 in the user's local storage 236 of FIG. 2.

Id., ¶[0314] (emphasis added).

If the user wants to post a message at step 712, the "create message" subprocess 714 is to be started with posting window 400 of FIG. 4, and then the "post message" subprocess 716 is also to be started **where the multimedia bookmark BBS server receives the message and stores it into the database of storage 214**. The detailed sub-processes of both "create message" and "post message" are described with reference to FIGS. 8 and 9, respectively.

Id., ¶[0299] (emphasis added).

When multimedia bookmark message 1004 is transferred to multimedia bookmark BBS server 1014, the included bookmark image data 1010 might be extracted from the transferred message and then stored as a separate file 1018 at the multimedia bookmark BBS server. In this case, multimedia bookmark image URI 1020 indicating the storage location of the extracted multimedia bookmark image file is added to multimedia bookmark section of the transferred message. Then, the modified message is **stored into the database of the multimedia bookmark BBS server**.

Id., ¶[0315] (emphasis added).

244. Kim describes that such editing commands from the remote administrator terminal (instructions to modify multimedia bookmarks) are sent to the multimedia bookmark BBS server when the administrator clicks on the save

button, and are thus received by the computer system from the remote administrator's client computer over the network (*e.g.*, the Internet). *Id.*, ¶¶[0335] (“After selecting the advertising multimedia bookmarks, the administrator can verify his/her selection by viewing a list of selected multimedia bookmark image 1832 and its multimedia bookmark information ... The administrator then completes the verification by **clicking on save button 1838.**”), [0336] (“And then, by clicking the **save button 1850** with “Advertising Multimedia Bookmark Mail” radio button 1846 checked, **the advertising multimedia bookmarks are stored on database** to be automatically attached into user's multimedia bookmark e-mail whenever a user sends a multimedia bookmark e-mail to other.”); *see also id.*, ¶¶[0314] (explaining that bookmarks are posted to the BBS server at the server side of the public network by clients), [0299] (explaining that bookmarks are stored to the BBS server database), [0315] (same).

245. A second instance of Kim's disclosure of receiving a cue point editing command over the network from a remote client terminal is where Kim incorporates by reference, Sull¹ (Exhibit 1007), which describes a computer server operably

¹ Sull (Application No. 09/911,293, and published as US 2002/0069218 (Exhibit 1007) is a prior published application to which Kim claims priority and which is incorporated by reference into Kim. Kim, ¶¶[0001], [0004].

connected to a network and connected to one or more client devices for adding and editing multimedia bookmarks.

The system of the present invention can be a **computer server that is operably connected to a network that has connected to it one or more client devices**. Local storage on the server can optionally include a database and sufficient circuitry and/or logic, in the form of hardware and/or software in any combination that facilitates the storing, indexing, searching, retrieving and/or rendering of **multimedia information**.

Sull, ¶[0055] (emphasis added).

The present invention provides a system and method for accessing multimedia content stored in a multimedia file having a beginning and an intermediate point, the content having at least one segment at the intermediate point. At a minimum, the system includes a **multimedia bookmark**, the multimedia bookmark **having content information about the segment** at the intermediate point, wherein a user can utilize the multimedia bookmark to access the segment without accessing the beginning of the multimedia file.

Sull, ¶[0051] (emphasis added).

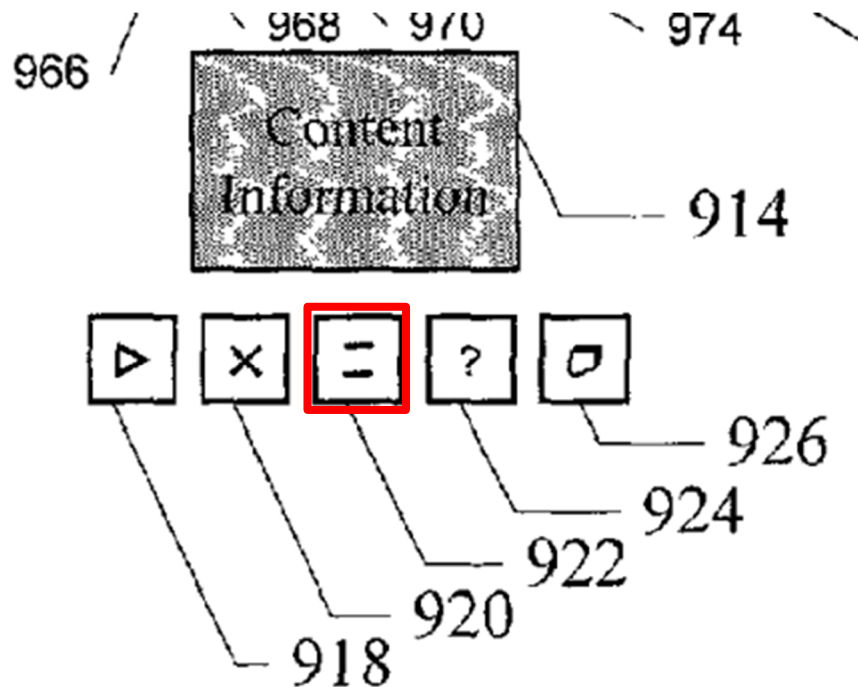
246. Kim, through Sull, discloses that to edit an existing bookmark, a user enters an edit mode by viewing a bookmark list 916 that visually displays a saved

bookmark and its content information, and then selecting the “add-bookmark-title control” button 922 to edit the title of the existing bookmark.

FIG. 9 shows an example of a user interface incorporating the multimedia bookmark of the present invention. The user interface 900 is composed of a playback area 912 and a **bookmark list 916**. ... Also, in the bookmark list 916, the saved bookmark is visually displayed with its content information. For example, a spatially reduced thumbnail image corresponding to the temporal location of interest saved by a user in the case of a multimedia bookmark is presented to help the user to easily recognize the previously bookmarked content of the video.

In the bookmark list 916, every bookmark has five bookmark controls just below its visually displayed content information. The left-most play-bookmark control button 918 is for playing a bookmarked multimedia content from a saved bookmarked position. The delete-bookmark control button 920 is for managing bookmarks. If this button is selected, the corresponding bookmark is deleted from the persistent storage. **The add-bookmark-title control button 922 is used to input a title of bookmark given by a user.** If this button is not selected, a default title is used. ...

Sull, ¶¶[0205]-[0206] (emphasis added).



Sull, FIG. 9 (excerpted and annotated, showing “add-bookmark-title” button for a bookmark).

247. The user is then prompted to enter a title and a check is made to determine whether the user entered a title, which results in a cue point editing command being sent to the computer server—as such, the cue point editing command is received when a new title is saved and the bookmark is thereafter “redisplayed with their content information, including the titles.”

FIG. 14 is a flow chart illustrating the process of adding a title to a multimedia bookmark. **When the add-bookmark-title control is selected** (step 1038 of FIG. 10), the program goes through this portion 1400 of the method of the present invention. In this routine, **the user will be prompted to enter a title** in step 1402 for the saved multimedia bookmark. **A check is made to**

determine if the user entered a title in step 1404. If not, the program may provide a default title in step 1406 that may be made in accordance with a predetermined routine. In any case, execution proceeds to step 1408, where the list of multimedia bookmarks is redisplayed with their content information, including the titles and bookmark controls. Thereafter, the method executes the clearing-off routine of step 1610 of FIG. 16.

Sull, ¶[0213] (emphasis added).

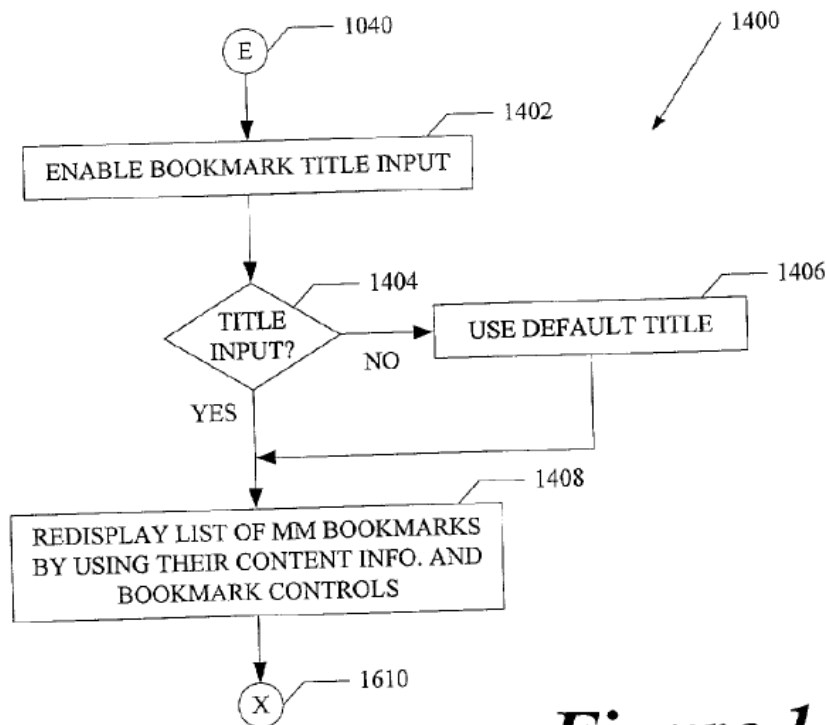


Figure 14

Sull, FIG. 14.

248. A POSITA would have understood the plain meaning of “cue point editing command” to be a command, sent from the remote client terminal and received by the computer system, to edit a portion of the cue point information. *See*

limitation [13.5] (indicating that “according to the cue point editing command” “a portion of the cue point information” is “modif[ied]”) (emphasis added). Further, a POSITA would have understood the word “command” (in “cue point editing command”) to mean an instruction to a computer program, due to how this word is used in the ’286 Patent.

249. The ’286 Patent describes that “the command, ‘RETRIEVE (‘VID—0001’), is used to retrieve a video file identified as ‘VID—0001’ from the video server 112. When the video server 112 receives the ‘RETRIEVE’ command, a relational database may be queried to find an entry for ‘VID—0001’ ... The video server 112 may then use various algorithms to select a download location and download the file.” ’286 Patent, 7:33-41 (emphasis added). In other words, an instruction is sent to the video server to retrieve a video, to which the video server responds. *See also id.*, 13:65-67 (describing “Visual Basic commands for an interpreter” that are instructions to the interpreter “to generate the GUI”) (emphasis added), 42:45-48 (describing a “command 1640” “transmitted by a client, received by a host or application” “requesting that a dataset is built for all ‘Exterior’ cue points” and the host or application responds to the instruction by “generat[ing] dataset 1650.”) (emphasis added).

250. This understanding, that a “command” is an instruction to a computer program, also comports with contemporaneous technical dictionaries, *e.g.*, the

Microsoft Computer Dictionary defines a “command” as “An instruction to a computer program that, when issued by the user, causes an action to be carried out.” Exhibit 1017.

251. Therefore, a POSITA would have understood that the user’s clicking the “save” button 1850 in Kim’s interface of FIG. 18C or selecting “add-bookmark-title” control button 922 in Sull’s interface of FIG. 9 results in sending a cue point editing command—an instruction to the computer system to edit a portion of the annotation—because the computer system responds by carrying out the requested action (*see* limitation [13.5] below).

252. To the extent Kim does not explicitly disclose receiving the editing command from a remote client terminal, a POSITA would have found it obvious from Kim and Sull’s teachings to use client 230 to send the editing command. For example, a POSITA would have appreciated from Kim’s and Sull’s client-server architecture that adding and managing bookmarks is functionality that exists at the client 230. Kim, ¶¶[0275], [0277], FIGs. 1-2; *see also* Sull, ¶[0055]. Therefore, a POSITA would have found it obvious to place the bookmark editing functionality at client 230 as editing bookmarks is similar to functions already handled by client 230.

253. Further, a POSITA would have appreciated that it was commonplace in bulletin board systems during Kim’s timeframe to have administrators be clients of the system with additional administrative privileges as compared to other clients.

Therefore, a POSITA would have found it obvious for the administrator in Kim to use client 230 to send editing commands to the servers of the computer system.

6. [13.5] modifying a portion of the cue point information at the server side of the public network according to the cue point editing command received over the public network from the client terminal;

254. In my opinion, Kim discloses limitation [13.5].

255. Kim discloses modifying a portion of the cue point information (e.g., editing multimedia bookmark information) at the server side of the public network (e.g., at multimedia bookmark BBS server 212) according to the cue point editing command received over the public network from the client terminal (e.g., command received in limitation [13.4]).

256. Kim teaches that when client 230 edits a bookmark's video title and duration or edits a bookmark's title (*see* my discussion above of limitation [13.4]), the edited information is sent over the network to modify the bookmark at multimedia bookmark BBS server 212 where it is saved in a database.

After verifying the advertising multimedia bookmarks, the manager can preview the selected advertising multimedia bookmarks. FIG. 18C illustrates an exemplary GUI screen to preview the advertising multimedia bookmarks selected by the administrator. The preview window 1840 is similar to the advertising multimedia bookmark part 1718 of the multimedia bookmark e-mail 1710 in FIG. 17. The administrator can verify

the final format of advertising multimedia bookmarks to be attached to a multimedia bookmark e-mail sent by a user. The advertising multimedia bookmark 1842 can be played by clicking on play button 1844. And then, by clicking the save button 1850 with "Advertising Multimedia Bookmark Mail" radio button 1846 checked, the advertising multimedia bookmarks are stored on database to be automatically attached into user's multimedia bookmark e-mail whenever a user sends a multimedia bookmark e-mail to other. **By clicking the save button 1850** with "Advertising News Letter" radio button 1848 checked, **the advertising multimedia bookmarks are stored on database** to be automatically attached into a news letter e-mail whenever a promotion is done by sending a news letter e-mail to users.

Kim, ¶[0336] (emphasis added).

If the user wants to post a message at step 712, the "create message" subprocess 714 is to be started with posting window 400 of FIG. 4, and then the "post message" subprocess 716 is also to be started where **the multimedia bookmark BBS server** receives the message and **stores it into the database of storage 214**. The detailed sub-processes of both "create message" and "post message" are described with reference to FIGS. 8 and 9, respectively.

Id., ¶[0299] (emphasis added).

When multimedia bookmark message 1004 is transferred to multimedia bookmark BBS server 1014, the included bookmark image data 1010 might be extracted from the transferred message and then stored as a separate file 1018 at the multimedia bookmark BBS server. In this case, multimedia bookmark image URI 1020 indicating the storage location of the extracted multimedia bookmark image file is added to multimedia bookmark section of the transferred message. Then, **the modified message is stored into the database of the multimedia bookmark BBS server.**

Id., ¶[0315] (emphasis added).

FIG. 14 is a flow chart illustrating the process of adding a title to a multimedia bookmark. When the add-bookmark-title control is selected (step 1038 of FIG. 10), the program goes through this portion 1400 of the method of the present invention. In this routine, the **user will be prompted to enter a title in step 1402 for the saved multimedia bookmark.** A check is made to determine if the user entered a title in step 1404. If not, the program may provide a default title in step 1406 that may be made in accordance with a predetermined routine. In any case, **execution proceeds to step 1408,** where the **list of multimedia bookmarks is redisplayed with their content information, including the titles and bookmark controls.** Thereafter, the method executes the clearing-off routine of step 1610 of FIG. 16.

Sull, ¶[0213] (emphasis added); *see also* ¶¶[0262]-[0263], FIG. 14.

7. **[13.6] receiving a cue point selection over the network from the remote client terminal identifying one of the cue points corresponding to a portion of the video information of one of the video files; and**

257. In my opinion, Kim discloses limitation [13.6].

258. Kim teaches multiple instances of receiving a cue point selection (*e.g.*, receiving request to play selected bookmark via play-bookmark control button 118 or play control button 612) over the network from the remote client terminal (*e.g.*, from client 230 over the Internet) that identifies one of the cue points corresponding to a portion of the video information of one of the video files (*e.g.*, selected multimedia bookmark identifies a video portion).

259. A first instance of Kim disclosing receiving a cue point selection is in the context of a user interface having a bookmark list 116 for a particular video where in the bookmark list 116, the saved bookmark is visually displayed with its content information. Under each bookmark is a set of controls which includes a “play-bookmark control button 118” that client 230 selects to play a bookmarked multimedia content from a saved bookmarked position.

FIG. 1 shows an exemplary GUI screen incorporating the multimedia bookmark of the previous art, that is, commonly-owned, copending U.S. patent application Ser. No. 09/911,293 filed Jul. 23, 2001 (Publication No. 2002/0069218), and additional features, according to an embodiment of the techniques of the present disclosure. The user interface of the

player window 102 is composed of a playback area 112 and a **bookmark list 116**. ... Also, **in the bookmark list 116, the saved bookmark is visually displayed with its content information**. For example, a spatially reduced image (or thumbnail image) corresponding to the temporal location of interest saved by a user in case of multimedia bookmark is presented to help the user to easily recognize the previously bookmarked content of the video.

In the bookmark list 116, which provides a personalization of the stored multimedia bookmarks, every bookmark has five bookmark controls just below its visually displayed content information. **The left-most play-bookmark control button 118 is for playing a bookmarked multimedia content from a saved bookmarked position**. ...

Kim, ¶¶[0275]-[0276] (emphasis added).

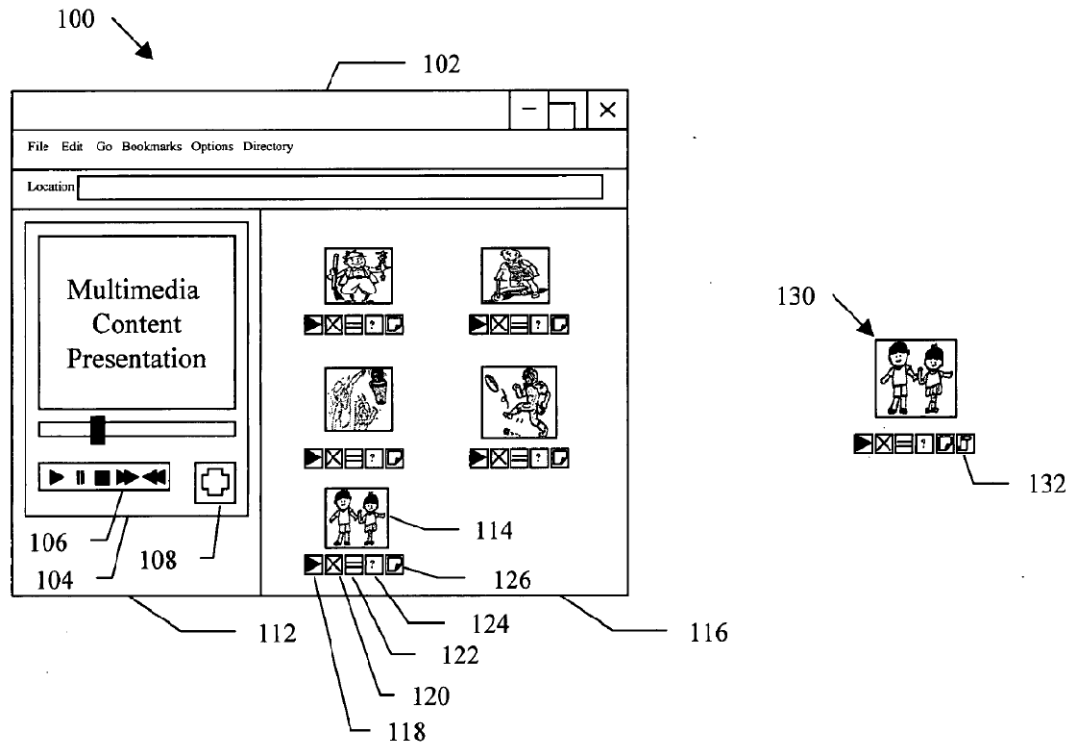


FIG. 1

Id., FIG. 1.

260. A second instance of Kim disclosing receiving a cue point selection is in the context of Kim's Fig. 3 interface that includes a message list window 300 showing a multimedia bookmark for each message, allowing a multimedia bookmark BBS users to easily identify a desired message. Kim teaches that client 230 selects a bookmark, such that message window 600 is displayed along with a play control button 612 which, when selected, causes video playback from the bookmarked position.

FIGS. 3, 4, 5, and 6 illustrate exemplary GUI screens of a multimedia bookmark BBS, according to an embodiment of the

present disclosure. **FIG. 3 illustrates an exemplary GUI screen of a message list window 300 of a multimedia bookmark BBS,** according to an embodiment of the present disclosure. In the figure, **message list window 300** of the multimedia bookmark BBS comprises general components of conventional BBS such as title of a message 312, uploading date of the message 314, and writer of the message 316. Furthermore, message list window 300 of the present disclosure **includes multimedia bookmark of the message 310 for each message.** By viewing the visual information of multimedia bookmarks, the multimedia bookmark BBS users **can easily identify a message in which they are interested.** The "write" (post or upload) control button 318 is selected when a user wants to post a multimedia bookmark. FIG. 4 shows the next GUI screen when the user selects the "write" control button.

Id., ¶[0290] (emphasis added).

After at least one multimedia bookmark message is posted to a multimedia bookmark BBS by a user, the user and the others can retrieve the message from the multimedia bookmark BBS. **FIG. 6 shows a message window 600 that is displayed when a message is selected from the message list window 300 of FIG. 3,** herein the selection is caused by clicking multimedia bookmark image 310 or title 312. FIG. 6 illustrates an exemplary GUI screen of a message window 600 of a multimedia bookmark BBS, according to an embodiment of the present disclosure. In the figure, message window 600 comprises multimedia

bookmark image 610, **play control button 612**, opener page control button 614, send-mail control button 616, textual description 618 for the video content from which the corresponding multimedia bookmark included in the selected message is captured, text box 620 for title of the selected message, and user description 622. **By selecting the play control button 612, the user can watch the video from the bookmarked position.** Note that the video will be played according to the predetermined duration set by a posting user or an administrator of multimedia bookmark BBS. ...

Id., ¶[0295] (emphasis added).

261. Kim also describes that the cue point selection is received over the network (e.g., Internet) from the client, as the client 230 sends a bookmark selection to the server over the network.

Media host 220 comprises streaming server 222, bookmark server 224 and storage 226 for archiving media files. Bookmark server 224 is responsible for handling the request from bookmark BBS server 212. The multimedia bookmark server obtains a bookmark image at the required position in accordance with the request and then sends the captured bookmark image to multimedia bookmark BBS server 212 as a reply for the request of multimedia bookmark image. **Streaming server 222 is responsible for the request from client 230 to play a video.**

Id., ¶[0289] (emphasis added).

FIG. 2 illustrates the general system architecture of the multimedia bookmark BBS, according to an embodiment of the present disclosure. The system comprises multimedia bookmark BBS server 212 located in web host 210, multimedia bookmark server 224 located in media host 220 and multimedia bookmark client 238 located in client 230. **The web host 210, media host 220 and client 230 are connected by a conventional communication network ("NETWORK").**

Id., ¶[0283] (emphasis added); *see also* ¶[0349].

262. In Kim, each bookmark (*i.e.*, cue point) corresponds to a portion (including a start time and duration) of a video.

Alternatively, the multimedia bookmark image might be obtained by multimedia bookmark client 238 that can capture and reduce a frame of video displayed in media player 234. The client 238, which is application software responsible for interactions between web browser 232 and local storage 236, stores the multimedia bookmark image into local storage 236 together with the bookmark information such as video URI, **start time, duration** and etc. The bookmark client 238 is also used to load the multimedia bookmark saved at the local storage into the web browser, so that the web browser can display the multimedia bookmark image and its information.

Id., ¶[0287] (emphasis added).

Then, with the posting window 400 in FIG. 4, the user selects and fills up other fields such as duration control box 414, title text input field 416 and description text input field 418. **The duration control box controls the allowable duration to play the multimedia bookmark from its bookmarked position. For example, the multimedia bookmark can be played for 30 seconds, 1 minutes, 2 minutes, 3 minutes or even to the end of video file.** Note that this duration can be set by an administrator of the multimedia bookmark BBS in order to limit or control the allowable duration of playing. Finally, the user can post the message including the selected multimedia bookmark by clicking on the submit control button 420.

Id., ¶[0293] (emphasis added).

FIG. 10 illustrates an exemplary structure of the multimedia bookmark message which is posted to multimedia bookmark BBS server 212 by client 230 of FIG. 2, according to an embodiment of the present disclosure. In the figure, the multimedia bookmark message 1004 in the client 1002 has a body section 1006 and multimedia bookmark section 1008. ... [T]he multimedia bookmark section 1008 **includes multimedia bookmark information such as** video URI, title page URI, **start time, duration**, and multimedia bookmark image data 1010. ...

Id., ¶[0314] (emphasis added); *see also* ¶[0316]-[0319].

8. [13.7] streaming the portion of the video information to the client terminal over the public network in response to receiving the cue point selection.

263. In my opinion, Kim discloses limitation [13.7].

264. Kim teaches streaming the portion of the video information to the client terminal (*e.g.*, streaming the portion of the video corresponding to the selected bookmark to client 230) over the public network (*e.g.*, the Internet) in response to receiving the cue point selection (*e.g.*, the bookmark selected via play-bookmark control button 118 or play control button 612 received in limitation [13.6]). *See my discussion above of limitation [13.6].*

265. Kim discloses that when client 230 selects a bookmark to playback, video streaming of the selected video corresponding to the bookmark begins from the start point and for the duration indicated by the bookmark.

Media host 220 comprises streaming server 222, bookmark server 224 and storage 226 for archiving media files. ... The multimedia bookmark server obtains a bookmark image at the required position in accordance with the request and then sends the captured bookmark image to multimedia bookmark BBS server 212 as a reply for the request of multimedia bookmark image. **Streaming** server 222 is responsible for the request from client 230 to play a video.

Kim, ¶[0289] (emphasis added).

After at least one multimedia bookmark message is posted to a multimedia bookmark BBS by a user, the user and the others can retrieve the message from the multimedia bookmark BBS. FIG. 6 shows a message window 600 that is displayed when a message is selected from the message list window 300 of FIG. 3, herein the selection is caused by clicking multimedia bookmark image 310 or title 312. FIG. 6 illustrates an exemplary GUI screen of a message window 600 of a multimedia bookmark BBS, according to an embodiment of the present disclosure. In the figure, message window 600 comprises multimedia bookmark image 610, play control button 612, opener page control button 614, send-mail control button 616, textual description 618 for the video content from which the corresponding multimedia bookmark included in the selected message is captured, text box 620 for title of the selected message, and user description 622. **By selecting the play control button 612, the user can watch the video from the bookmarked position. ...**

Id., ¶[0295] (emphasis added).

Web host 210 provides lists of videos which have corresponding links or URIs of the videos stored at media host 220 with a hypertext markup language (HTML). Media host 220 stores the videos in its local storage 226, and provides them to client 230 when they are requested. Client 230 can select one of the videos from the lists displayed in web browser 232. The selection process requests the video to be serviced to media host 220, and

client 230 receives the video streamed by streaming server 222 of media host 220 and displays the video on media player 234.

Id., ¶[0284] (emphasis added).

FIG. 12 is an exemplary flowchart illustrating the method of reading a multimedia bookmark message from a multimedia bookmark BBS, according to an embodiment of the present disclosure. When the decision to read a message is made at step 706 of FIG. 7, the subprocess "read message" 708 of FIG. 7 starts at step 1202 of FIG. 12. The subprocess displays a message window 600 of FIG. 6 at step 1204. It then moves to decision step 1206 whether a user will play the bookmarked video segment in the message window or not. If the decision to play the bookmarked video segment is made at steps 1206, the subprocess moves to the "Multimedia bookmark play" subprocess at step 1210 which is also illustrated in more details in FIG. 13. ...

FIG. 13 is an exemplary flowchart illustrating the method of playing a multimedia bookmark from a multimedia bookmark BBS, according to an embodiment of the present disclosure. ... The subprocess then moves to step 1304 where the player window such as multimedia bookmark player 102 of FIG. 1 is opened and an additional browsing window might be opened, which displays a HTML page associated with the multimedia bookmark information such as the title page of the video. Within the player window, the video starts to play from the bookmarked position of the multimedia bookmark

information at step 1306. As used herein, playing "from" the bookmarked position means starting playback of the video from a frame at or near (typically, within a few seconds of) the bookmarked position.

Id., ¶¶[0318]-[0319] (emphasis added).

C. Claim 14

266. Claim 14 recites:

14. The method of claim 13, further comprising providing a cue point editing mode as part of the video editing environment configured to perform at least one of creating, modifying, deleting, or interpreting at least a portion of the set of cue point information according to the cue point editing command received over the public network from the client terminal.

267. In my opinion, Claim 14 is obvious over Kim.

268. Kim discloses the limitations recited in Claim 14, and discloses multiple instances of providing a cue point editing mode as part of the video editing environment (*e.g.*, interfaces and program modules for editing bookmarks provided by web host 210 and media host 220 and their servers and storages) configured to perform at least one of creating, modifying, deleting, or interpreting at least a portion of the set of cue point information (*e.g.*, editing multimedia bookmark) according to the cue point editing command received over the public network from the client terminal (*e.g.*, the command received in limitation [13.4]).

269. A first instance of Kim disclosing the limitations of Claim 14 is where Kim discloses a cue point editing mode being entered when, while in the interface of FIG. 18A, an administrator at client 230 clicks on a save button 1822 to select advertising bookmarks and enter an edit mode where a “GUI screen” (as shown in FIG. 18B) is displayed to edit video title or allowable playing duration in information field 1836 for the selected bookmarks. *See* my discussion above of limitation [13.4]; *see also*:

To prepare the advertising multimedia bookmarks, the administrator makes his/her own multimedia bookmarks from new or existing videos that he/she wants to advertise, and then selects some multimedia bookmarks to be attached from his/her own multimedia bookmarks. FIGS. 18A, 18B and 18C illustrate exemplary **GUI screens** of a managing tool for the **administrator to select the advertising multimedia bookmarks from his/her own multimedia bookmarks**, according to an embodiment of the present disclosure. FIG. 18A illustrates an exemplary GUI screen to select advertising multimedia bookmarks from the administrator's own multimedia bookmarks. ... By checking the check box 1820 below an interesting multimedia bookmark 1818, the administrator selects one advertising multimedia bookmark. ... The administrator then completes the selection by **clicking on save button 1822**, and a new GUI screen 1830 of FIG. 18B will appear.

FIG. 18B illustrates an exemplary **GUI screen** to list the advertising multimedia bookmarks selected by the administrator. After selecting the advertising multimedia bookmarks, the administrator can verify his/her selection by viewing a list of selected multimedia bookmark image 1832 and its multimedia bookmark information described in information fields 1836 such as video title, file location/name, start-time, duration, and the related Web page URI. The **administrator** can **edit video title or allowable playing duration in information field 1836**. Also, the administrator can remove the multimedia bookmark 1832 from the selected list by clicking on delete button 1834. The administrator then completes the verification by clicking on save button 1838, and a new GUI screen 1840 of FIG. 18C will appear.

Kim, ¶¶[0334]-[0335] (emphasis added); *see also* ¶[0336].

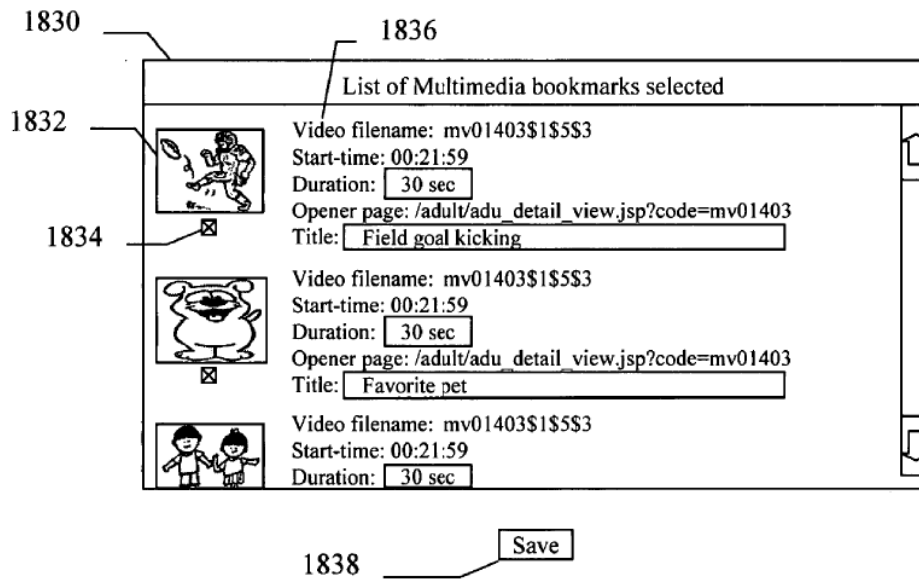


FIG. 18B

Id., FIG. 18B.

270. A second instance of Kim disclosing the limitations of Claim 14 is where Sull (incorporated by reference in Kim, as I discuss above for limitation [13.4]) describes entering a cue point editing mode when client 230 selects an “add-bookmark-title” control button 922 to edit an existing bookmark’s title and enters an edit mode that prompts the user to enter a bookmark title.

FIG. 14 is a flow chart illustrating the process of adding a title to a multimedia bookmark. **When the add-bookmark-title control is selected** (step 1038 of FIG. 10), the program goes through this portion 1400 of the method of the present invention. In this routine, **the user will be prompted to enter a title** in step 1402 for the saved multimedia bookmark. **A check is made to determine if the user entered a title in step 1404.** If not, the

program may provide a default title in step 1406 that may be made in accordance with a predetermined routine. In any case, execution proceeds to step 1408, where the list of multimedia bookmarks is redisplayed with their content information, including the titles and bookmark controls. Thereafter, the method executes the clearing-off routine of step 1610 of FIG. 16.

Sull, ¶[0213] (emphasis added).

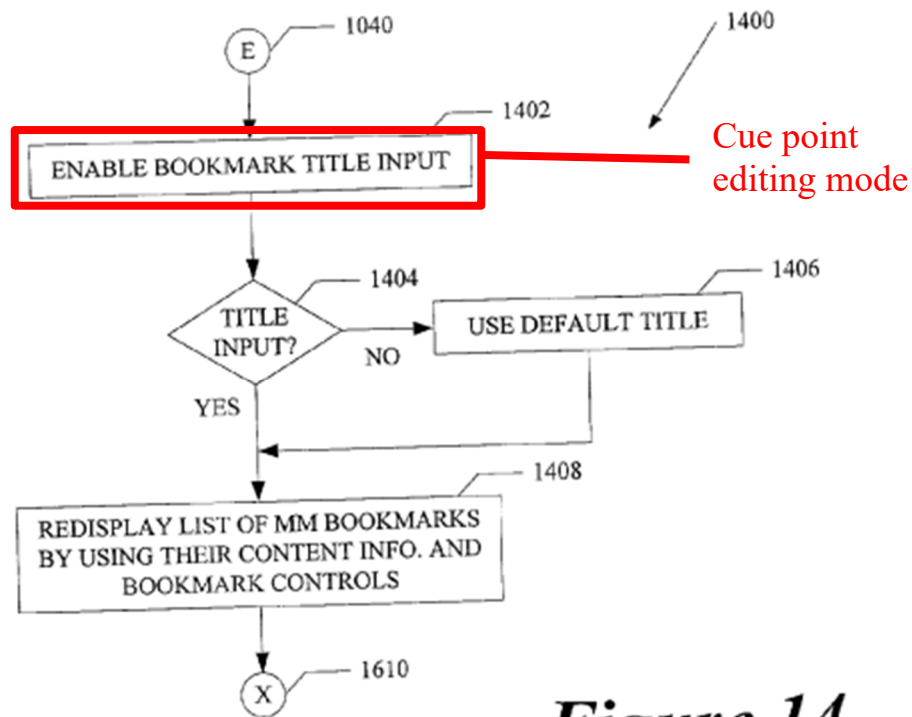


Figure 14

Sull, FIG. 14 (annotated).

271. Kim's multimedia bookmark BBS server 212 modifies at least a portion of the bookmark information according to the editing command received over the Internet from client 230. *See my discussion above of [13.4]-[13.5]; see also:*

After verifying the advertising multimedia bookmarks, the manager can preview the selected advertising multimedia bookmarks. FIG. 18C illustrates an exemplary GUI screen to preview the advertising multimedia bookmarks selected by the administrator. The preview window 1840 is similar to the advertising multimedia bookmark part 1718 of the multimedia bookmark e-mail 1710 in FIG. 17. The administrator can verify the final format of advertising multimedia bookmarks to be attached to a multimedia bookmark e-mail sent by a user. ... And then, **by clicking the save button 1850** with "Advertising Multimedia Bookmark Mail" radio button 1846 checked, the **advertising multimedia bookmarks are stored on database** to be automatically attached into user's multimedia bookmark e-mail whenever a user sends a multimedia bookmark e-mail to other. **By clicking the save button 1850** with "Advertising News Letter" radio button 1848 checked, the **advertising multimedia bookmarks are stored on database** to be automatically attached into a news letter e-mail whenever a promotion is done by sending a news letter e-mail to users.

Kim, ¶[0336] (emphasis added); *see also* ¶[0335].

FIG. 14 is a flow chart illustrating the process of adding a title to a multimedia bookmark. **When the add-bookmark-title control is selected** (step 1038 of FIG. 10), the program goes through this portion 1400 of the method of the present invention. In this routine, **the user will be prompted to enter a title** in step 1402 for the saved multimedia bookmark. **A check is made to**

determine if the user entered a title in step 1404. If not, the program may provide a default title in step 1406 that may be made in accordance with a predetermined routine. In any case, **execution proceeds to step 1408, where the list of multimedia bookmarks is redisplayed with their content information, including the titles and bookmark controls.** Thereafter, the method executes the clearing-off routine of step 1610 of FIG. 16.

Sull, ¶[0213] (emphasis added); *see also* ¶¶[0205]-[0206].

D. Claim 15

272. Claim 15 recites:

15. The method of claim 14, further comprising: receiving an edit mode entry signal over the network from the client terminal representing entry into the cue point edit mode, and switching the video editing environment to the cue point editing mode in response to receiving the edit mode entry signal.

273. In my opinion, Claim 15 is obvious over Kim.

274. Kim discloses the limitations recited in Claim 15, as Kim teaches receiving an edit mode entry signal (*e.g.*, signal for entry into the interface(s) to edit a bookmark, from selecting bookmarks to edit by clicking on Kim's save button 1822 or Sull's add-bookmark-title control button 922, as I discuss above for Claim 14) over the network from the client terminal (*e.g.*, from client 230 over the Internet) representing entry into the interface(s) for editing bookmarks and switching the video editing environment to the cue point editing mode in response to receiving the

edit mode entry signal (e.g., displaying Kim's FIG. 18B interface for editing a bookmark's video title or allowable playing duration, or Sull's FIG. 14 interface for editing a bookmark's title). See my discussion above of limitation [13.4] and claim 14; *see also*:

To prepare the advertising multimedia bookmarks, the administrator makes his/her own multimedia bookmarks from new or existing videos that he/she wants to advertise, and then selects some multimedia bookmarks to be attached from his/her own multimedia bookmarks. FIGS. 18A, 18B and 18C illustrate exemplary GUI screens of a managing tool for the **administrator to select the advertising multimedia bookmarks from his/her own multimedia bookmarks**, according to an embodiment of the present disclosure. FIG. 18A illustrates an exemplary GUI screen to select advertising multimedia bookmarks from the administrator's own multimedia bookmarks. ... The administrator then completes the selection by **clicking on save button 1822**, and a new GUI screen 1830 of FIG. 18B will appear.

FIG. 18B illustrates an exemplary GUI screen to list the advertising multimedia bookmarks selected by the administrator. After selecting the advertising multimedia bookmarks, the administrator can verify his/her selection by viewing a list of selected multimedia bookmark image 1832 and its multimedia bookmark information described in information fields 1836 such as video title, file location/name, start-time, duration, and the

related Web page URI. The **administrator** can **edit video title or allowable playing duration in information field 1836**. Also, the administrator can remove the multimedia bookmark 1832 from the selected list by clicking on delete button 1834. The administrator then completes the verification by clicking on save button 1838, and a new GUI screen 1840 of FIG. 18C will appear.

Kim, ¶¶[0334]-[0335] (emphasis added); *see also* ¶[0336], FIG. 2 (showing a web host 210, web server 216, and web browser 230).

In the bookmark list 916, every bookmark has five bookmark controls just below its visually displayed content information. The left-most play-bookmark control button 918 is for playing a bookmarked multimedia content from a saved bookmarked position. The delete-bookmark control button 920 is for managing bookmarks. If this button is selected, the corresponding bookmark is deleted from the persistent storage. **The add-bookmark-title control button 922 is used to input a title of bookmark given by a user.** If this button is not selected, a default title is used. ...

Sull, ¶[0206] (emphasis added), *see also* ¶[0213].

275. A POSITA would have understood that this change from one interface to another would be based on a request (e.g., HTTP request) from client 230 to the web server, as such request/response sequence is the manner in which a HTML page is requested by a client and provided by a web server over the Internet.

276. A POSITA would have understood the plain meaning of “edit mode entry signal” to be a signal, sent from the remote client terminal and received by the computer system, to enter an edit mode. Further, a POSITA would have understood the word “signal” (in “edit mode entry signal”) to mean an indication. *See* Exhibit 1018 (“Signal.” Merriam-Webster.com Dictionary, Merriam-Webster, <https://www.merriam-webster.com/dictionary/signal> (accessed March 29, 2024) (defining “signal” as “1: SIGN, INDICATION”)).

277. As the ’286 Patent explains, “in some embodiments, cue point edit mode will always be available, while other embodiments may require **active entry into that mode, for example by polling for input.**” ’286 Patent, 15:28-31 (emphasis added). The “input” may be “a particular event from an input device, like a mouse or keyboard.” *Id.*, 15:31-33; *see also id.*, 18:42:49 (explaining “the client may enter the cue point edit mode in any useful way” including “click[ing] on a video image” or “select[ing] a GUI control dedicated to providing entry into the cue point edit mode” or “select[ing] a portion of the video progress bar”).

278. Therefore, a POSITA would have understood that when a user clicking on Kim’s save button 1822 or Sull’s add-bookmark-title control button 922 for a selected bookmark, web host 210 receives an edit mode entry signal—an indication to the computer system to enter an edit mode.

E. Claim 18

279. Claim 18 recites:

18. The method of claim 13, further comprising receiving, over the network, authentication information relating to access authority of the remote client terminal.

280. In my opinion, Claim 18 is obvious over Kim.

281. Kim discloses the limitations recited in Claim 18, as Kim teaches receiving authentication information (*e.g.*, “information on the user” for accessing user’s DRM license key) relating to access authority of the remote client terminal (*e.g.*, dictating whether a digital rights management (“DRM”) packaged video can be accessed by a client 230). The bookmark server 2024 receives the authentication information (the “information on the user” noted in ¶[0345] of Kim) over the network interconnecting bookmark server 2024 and the client, and the bookmark server 2024 uses that received information to retrieve the user’s license key from database 2014 to access DRM packaged video.

When a user of client 2026 makes a multimedia bookmark while playing a video with media player 2028, the client requests remote multimedia bookmark server 2024 to capture a multimedia bookmark image from the video **with information on the user**. Then, before capturing the multimedia bookmark image from the video, **multimedia bookmark server 2024 negotiates with license server 2016** and Web server 2018, and **requests license server 2016 to retrieve a license key of the**

user from database 2014. The license server will then return the license key of the user to the multimedia bookmark server if it exists. The multimedia bookmark server then unpacks the requested DRM packaged video stored at storage 2022 with the returned license key of the user, and captures a multimedia bookmark image from the video at a requested bookmarked position. The extracted multimedia bookmark image is sent back to client 2026.

Id., ¶[0345] (emphasis added).

For some systems where only authorized users are allowed to access videos, the videos can be packaged with digital rights management (DRM) technologies. For the systems, making multimedia bookmarks on the DRM packaged videos needs more sophisticated controls. FIGS. 20A and 20B illustrate the general system architectures for making multimedia bookmarks on the DRM packaged videos when multimedia bookmark images are captured at a remote host or client computer itself, respectively, according to an embodiment of the present disclosure.

FIG. 20A illustrates the general system architecture for making multimedia bookmarks on the DRM packaged videos, wherein multimedia bookmark server module 2024 is running at a remote host computer. In the figure, video encoder 2010 encodes and packages the video source 2012 with DRM. The DRM packaged video is stored at storage 2022 where the packaged videos are accessed by streaming server 2020 and multimedia bookmark

server 2024. A license key used to unpack the packaged video is stored at database 2014 of license server 2016. The Web server 2018 also has the information related to the license key and users, which is required by the client 2026 when the video starts to be played. Client 2026 comprises media player 2028 and multimedia bookmark client 2030 that takes charge of making and managing multimedia bookmarks stored at local storage 2032.

Kim, ¶¶[0343]-[0344] (emphasis added).

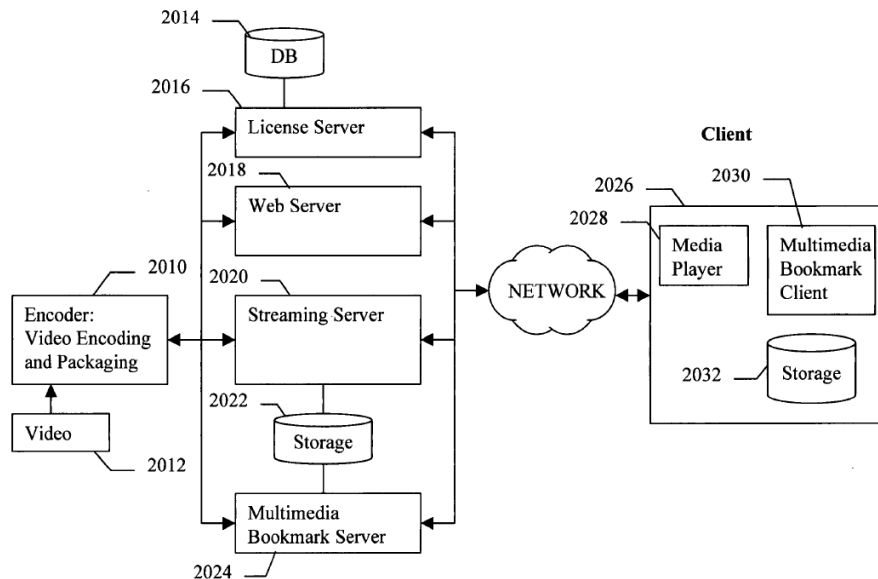


FIG. 20A

Id., FIG. 20A.

F. Claim 19

282. Claim 19 recites:

19. The method of claim 18, further comprising denying or allowing access to all or part of the video editing environment

based at least in part on the access authority of the remote client terminal.

283. In my opinion, Claim 19 is obvious over Kim.

284. Kim teaches denying or allowing access to all or part of the video editing environment (*e.g.*, denying access by a client 230 to part of the software executed by web host 210 and media host 220; denying access to bookmark a DRM packaged video) based at least in part on the access authority of the remote client terminal, as only client with the proper license keys may add/edit bookmarks for DRM packaged video.

285. Kim explains that when a user is establishing a multimedia bookmark while playing a DRM packaged video, and before capturing a multimedia bookmark image from the video, the multimedia bookmark server 2024 negotiates with the license server 2016 and the Web server 2018, and sends a request to the license server 2016 to retrieve a license key of the user from database 2014, if one exists.

When a user of client 2026 makes a multimedia bookmark while playing a video with media player 2028, the client requests remote multimedia bookmark server 2024 to capture a multimedia bookmark image from the video with information on the user. Then, before capturing the multimedia bookmark image from the video, multimedia bookmark server 2024 negotiates with license server 2016 and Web server 2018, and **requests license server 2016 to retrieve a license key of the user from**

database 2014. The license server will then return the license key of the user to the multimedia bookmark server if it exists. The multimedia bookmark server then unpacks the requested DRM packaged video stored at storage 2022 with the returned license key of the user, and captures a multimedia bookmark image from the video at a requested bookmarked position. The extracted multimedia bookmark image is sent back to client 2026.

Kim, ¶[0345] (emphasis added).

286. A POSITA would have understood that because the license key is required to capture the multimedia bookmark image, the multimedia bookmark server 2024 denies or allows access to all or part of the video editing environment based on whether the client 230 has a legitimate license key. Indeed, Kim discloses that the purpose of DRM protection for a video is to deny access to the video to a client who lacks the proper license key.

For some systems where only authorized users are allowed to access videos, the videos can be packaged with digital rights management (DRM) technologies. ...

Kim, ¶[0343] (emphasis added).

G. Claim 20

287. Claim 20 recites:

The method of claim 13, wherein the server video data is streamed progressively to a video playback environment at the remote client terminal, allowing the video player to begin displaying streamed server video data while remaining server video data continues to be streamed.

288. In my opinion, Claim 20 is obvious over Kim.

289. Kim discloses server video data being streamed progressively to a video playback environment (*e.g.*, streaming video progressively to playback area 112) at the remote client terminal (*e.g.*, client 230), allowing the video player (*e.g.*, multimedia player 104) to begin displaying streamed server video data while remaining server video data continues to be streamed (*e.g.*, client 230 renders streamed video data as it is received from streaming server 222). *See* my discussion above of limitation [13.7]; *see also*:

FIG. 1 shows an exemplary GUI screen incorporating the multimedia bookmark of the previous art, that is, commonly-owned, copending U.S. patent application Ser. No. 09/911,293 filed Jul. 23, 2001 (Publication No. 2002/0069218), and additional features, according to an embodiment of the techniques of the present disclosure. **The user interface of the player window 102 is composed of a playback area 112** and a bookmark list 116. Further, the playback area 112 **includes a multimedia player 104**. ...

Kim, ¶[0275] (emphasis added).

FIG. 13 is an exemplary flowchart illustrating the method of playing a multimedia bookmark from a multimedia bookmark BBS, according to an embodiment of the present disclosure. When the decision to **play a bookmark** is made at step 1106 or 1206 of FIG. 11 or 12, the subprocess "Multimedia bookmark play" 1110 or 1210 of FIG. 11 or 12 starts at step 1302 of FIG. 13. The subprocess then moves to step 1304 where the **player window such as multimedia bookmark player 102** of FIG. 1 **is opened** and an additional browsing window might be opened, which displays a HTML page associated with the multimedia bookmark information such as the title page of the video. **Within the player window, the video starts to play from the bookmarked position of the multimedia bookmark information at step 1306.** ...

Id., ¶[0319] (emphasis added).

Web host 210 provides lists of videos which have corresponding links or URIs of the videos stored at media host 220 with a hypertext markup language (HTML). Media host 220 stores the videos in its local storage 226, and provides them to client 230 when they are requested. Client 230 can select one of the videos from the lists displayed in web browser 232. The selection process requests the video to be serviced to media host 220, and **client 230 receives** the video **streamed by streaming server 222** of media host 220 and **displays the video on media player 234.**

Id., ¶[0284] (emphasis added).

A POSITA would have understood the word “progressively” (in “streamed progressively”) to mean streaming that allows display of the streamed server video data to begin while the remaining server video data is still being streamed, based on both the claim language and the ’286 Patent. *See* claim 20 (“streaming progressively ... allowing the video player to begin displaying streamed server video data while remaining server video data continues to be streamed.”); ’286 Patent 25:41-47 (“Further, video files may load in various ways, including as a complete file, streaming, or progressive. Progressive loading of video files, for example, allows the video frame to begin playing the portion of video data which has already loaded while the remainder of video data continues to load.”), 4:7-10 (“client video data is transmitted progressively, allowing the video player to begin displaying transmitted client video data while remaining client video data continues to be transmitted.”).

XIII. GROUND 4: CLAIMS 13-15 AND 18-20 ARE OBVIOUS OVER KIM AND GUPTA

A. Claim 13

290. Claim 13 recites:

[13.P] 13. A method for handling video cue points, the method comprising:

[13.1] providing a computer system having at least one computer-readable medium comprising a video editing environment, and being disposed at a server side of a public network;

[13.2] receiving a set of video information at the video editing environment using the computer system;

[13.3] receiving a set of cue point information at the video editing environment using the computer system;

[13.4] receiving a cue point editing command over the network from a remote client terminal disposed at a client side of the public network using the computer system;

[13.5] modifying a portion of the cue point information at the server side of the public network according to the cue point editing command received over the public network from the client terminal;

[13.6] receiving a cue point selection over the network from the remote client terminal identifying one of the cue points corresponding to a portion of the video information of one of the video files; and

[13.7] streaming the portion of the video information to the client terminal over the public network in response to receiving the cue point selection.

291. As explained above in Ground 3, it is my opinion that Kim teaches all limitations of claim 13. To the extent it is argued that Kim alone fails to disclose or render obvious receiving a cue point editing command over the network from a remote client terminal of limitation [13.4], it is my opinion that a POSITA would

have nonetheless found it obvious to incorporate such functionality into Kim in light of Gupta such that claim 13 is rendered obvious by Kim in view of Gupta.

1. **[13.4] receiving a cue point editing command over the network from a remote client terminal disposed at a client side of the public network using the computer system;**

292. Kim discloses receiving a cue point editing command from an administrator. To the extent that Kim alone does not suggest or render obvious the administrator (or other editor user) operating from a remote client terminal disposed at a client side of a public network, Gupta discloses this aspect. *See* Section XII.B.5 ([13.4]), *supra*.

293. A POSITA would have been motivated to implement an “edit” option on the client interface to allow the user to send annotation (cue point) editing commands to modify existing annotations, as taught by Gupta, in the method of Kim in order to allow users, including ones at the client side of the client-server arrangement, to modify existing bookmarks (cue points) to, for example, make the beginning and ending times more precise or change the title (as Kim already discloses an administrator has access to). A POSITA would have appreciated that providing remote access to the editing capability, as taught in Gupta, would have provided system users with increased level of access without requiring them to be physically present at the server (*e.g.*, an administrator would be able to edit annotations from anywhere in the world, not only a single centralized server). A

POSITA would have had a reasonable expectation of success with such a combination because Kim and Gupta are drawn to similar systems of adding bookmarks and annotations to portions of video. *Compare* Gupta, Abstract, ¶¶[0003], [0006]-[0007] with Kim, Abstract, ¶¶[0205]-[0209]. Further, a POSITA would have had a reasonable expectation of success with such a modification because it would have been a simple matter of programming the web host to accept commands not just from an interface on the server, but from the same or similar interface on a client. Thus, upon a command from the client computer after selecting the “edit” option, enter an “editing mode” and receive user commands to edit existing annotations and bookmarks. Such a modification would have been a simple and straightforward modification for a POSITA because Kim already discloses an administrator user accessing a management tool GUI to edit bookmarks and such program modules could be largely reused to provide this functionality at the client. Thus, Kim in view of Gupta teaches receiving a cue point editing command over the network from a remote client terminal disposed at a client side of a public network.

B. Claim 14

294. Claim 14 recites:

14. The method of claim 13, further comprising providing a cue point editing mode as part of the video editing environment configured to perform at least one of creating, modifying, deleting, or interpreting at least a portion of the set of cue point

information according to the cue point editing command received over the public network from the client terminal.

295. In my opinion, Claim 14 is obvious over Kim in view of Gupta.

296. As explained above with respect to claim 13, Kim as modified by Gupta teaches providing a cue point editing mode as part of the video editing environment to perform at least one of creating, modifying, deleting, or interpreting at least a portion of the set of cue point information according to the cue point editing command received over the public network from the client. *See* [13.4], *supra*.

C. Claim 15

297. Claim 15 recites:

15. The method of claim 14, further comprising: receiving an edit mode entry signal over the network from the client terminal representing entry into the cue point edit mode, and switching the video editing environment to the cue point editing mode in response to receiving the edit mode entry signal.

298. In my opinion, Claim 15 is rendered obvious by Kim in view of Gupta.

299. As explained above with respect to claim 13, Kim as modified by Gupta teaches receiving an edit mode entry signal (command to enter the “edit” option) over the network from the client terminal representing entry into the cue point edit mode, and switching the video editing environment to the cue point editing mode in response to receiving the edit mode entry signal. *See* [13.4], *supra*; Gupta, ¶¶[0044], [0047], [0082], [0124]-[0128].

D. Claims 18-20

300. Claim 18 recites:

18. The method of claim 13, further comprising receiving, over the network, authentication information relating to access authority of the remote client terminal.

301. Claim 19 recites:

19. The method of claim 18, further comprising denying or allowing access to all or part of the video editing environment based at least in part on the access authority of the remote client terminal.

302. Claim 20 recites:

The method of claim 13, wherein the server video data is streamed progressively to a video playback environment at the remote client terminal, allowing the video player to begin displaying streamed server video data while remaining server video data continues to be streamed.

303. As discussed above for Ground 3, Kim discloses the limitations of Claim 18-20. Therefore, in my opinion, these claims are likewise obvious over Kim and Gupta.

XIV. GROUND 5: CLAIMS 13-15 AND 18-20 ARE OBVIOUS OVER VASILEVSKY AND LOGAN

A. Summary of Vasilevsky

304. Vasilevsky (Exhibit 1008) is a publication of an international patent application, published on August 14, 2003. I understand that Vasilevsky constitutes prior art at least under pre-AIA 35 U.S.C. §§ 102(a), (b), and (e) even under the alleged May 7, 2006 priority date.

305. Vasilevsky describes a centralized Digital Video Recording and reproduction system with centralized media server allowing users to establish bookmarks via a graphical user interface display. Vasilevsky, Abstract. The media server is connected to client device receivers over a Home Area Network. *Id.*, 10:11-11:15. A user in this system can pause video playback at its client receiver device and create and/or modify bookmarks in the video program. *Id.*, 19:12-20:6, 21:9-22:15. A user may later access the created/modified bookmarks, and select a bookmark to resume playback at the bookmarked point. *Id.*, 20:3-6.

B. Summary of Logan

306. Logan (Exhibit 1009) is a publication of a U.S. patent application, published on May 15, 2003. I understand that Logan constitutes prior art at least under pre-AIA 35 U.S.C. §§ 102(a), (b), and (e) even under the alleged May 7, 2006 priority date.

307. Logan describes a system having a client-server architecture in which a client station can create or modify metadata, including tags and bookmarks—this metadata is sent to a remote station for sharing with other users. Logan, ¶¶[0015], [0048], [0049], [0051], [0053], [0057], [0065], [0112], [0124], [0134], [0221]-[0229], [0308], FIGs. 1-2. Logan teaches users creating and saving bookmarks as they watch video programming, in order to “save a segment” for watching later. *Id.*, ¶¶[0177]-[0179]. Logan also discloses that its system may be implemented over the public Internet. *Id.*, ¶¶[0050]-[0051].

C. Claim 13

308. Claim 13 recites:

[13.P] 13. A method for handling video cue points, the method comprising:

[13.1] providing a computer system having at least one computer-readable medium comprising a video editing environment, and being disposed at a server side of a public network;

[13.2] receiving a set of video information at the video editing environment using the computer system;

[13.3] receiving a set of cue point information at the video editing environment using the computer system;

[13.4] receiving a cue point editing command over the network from a remote client terminal disposed at a client side of the public network using the computer system;

[13.5] modifying a portion of the cue point information at the server side of the public network according to the cue point editing command received over the public network from the client terminal;

[13.6] receiving a cue point selection over the network from the remote client terminal identifying one of the cue points corresponding to a portion of the video information of one of the video files; and

[13.7] streaming the portion of the video information to the client terminal over the public network in response to receiving the cue point selection.

309. In my opinion, Claim 13 is obvious over Vasilevsky and Logan.

1. [13.P] A method for handling video cue points, the method comprising:

310. In my opinion, to the extent the preamble is limiting, Vasilevsky discloses limitation [13.P].

311. Vasilevsky describes a centralized Digital Video Recording (DVR) and reproduction system with centralized media server that allows users to pause and resume the same programs from the same marker points (*i.e.* “bookmarks”) established by any system user. Vasilevsky discloses the handling of video cue

points, as Vasilevsky discloses the system's "establish[ing]", "show[ing]", and "edit[ing]" of bookmarks.

A **centralized Digital Video Recording (DVR) and reproduction system (100)** linking several reproduction and control units (such as television receivers (124, 128, 132), etc.) to a centralized media server (152) allows system users (124, 128, 132) to not only access the same program or programs from different receivers (106, 110), but to pause and **resume the same programs from the same marker points ("bookmarks") established** by any **system user (124, 128, 132)**. The program "bookmark" representing the program pause point is conveniently **shown** via a graphical user interface display with a bookmark icon, including the name of the user (124, 128, 132) who established the bookmark and the exact pause point in the program.

Vasilevsky, Abstract (emphasis added).

The system of Claim 1, wherein said media server is adapted to enable any program bookmark to be **edited** by any authorized system user.

Id., claim 4 (emphasis added).

312. Vasilevsky's bookmarks are cue points, as they are electronic markers that are saved to memory which indicate exactly where the program was paused and

other information such as the identity of the user that created the bookmark and that date and time of the bookmark.

When a system user pauses the playback of a particular program, he or she can decide whether to **"bookmark"** the program. That is, an **electronic marker** can be **placed in memory indicating exactly where the program was paused**, along with **other particulars** such as the **identity of the user**, and the **date and time of the bookmark**. When a command is received to enter a bookmark, the Application Unit 134 causes the bookmark to be stored in a bookmark database 162, along with program identity information so that the bookmark can be matched to the appropriate program stored in memory 118.

Id., 19:12-20 (emphasis added).

313. Similarly, the '286 Patent describes "cue points" as indexed time positions or tags anywhere within a video file. The '286 Patent also describes that that "cue points" may include "meta-tags" that describe the information in the video and include descriptive identifiers.

In some embodiments, cue points may refer to **indexed time positions or tags anywhere within a video file**.

'286 Patent, 7:50-52 (emphasis added).

In yet other embodiments, cue points may be **video meta-tags**. Meta-tags may **describe the information in the video file**, rather than pointing to specific data locations within the file. For

example, video meta-tags for a video about a hotel Suite in downtown Denver may include **descriptive identifiers**, like “hotel,” “suite,” “kitchenette,” “luxury,” “downtown,” etc.

Id., 8:15-20 (emphasis added).

314. Therefore, Vasilevsky’s bookmarks are “cue points” in the context of the ’286 Patent claims.

2. [13.1] providing a computer system having at least one computer-readable medium comprising a video editing environment, and being disposed at a server side of a public network;

315. In my opinion, Vasilevsky in view of Logan teaches limitation [13.1].

316. Vasilevsky describes providing a computer system with servers and storage devices at a server side of a public network, and including a centralized media server comprising at least Video Decoder 120, Graphics Blender 122, Application Unit 134, Virtual Frame Buffer 136, Application Services Unit 152, storages Video Manager 116, Memory 118, and Bookmark Database 162.

A centralized Digital Video Recording (DVR) and reproduction system (100) linking several reproduction and control units (such as television receivers (124, 128, 132), etc.) to **a centralized media server (152)** allows system users (124, 128, 132) to not only access the same program or programs from different receivers (106, 110), but to **pause and resume the same programs from the same marker points ('bookmarks')** established by any system user (124, 128, 132). The program

"bookmark" representing the program pause point is conveniently shown via a graphical user interface display with a bookmark icon, including the name of the user (124, 128, 132) **who established the bookmark and the exact pause point in the program.**

Vasilevsky, Abstract (emphases added).

... the components 120, 122, 134, 136 and 152 are part of the centralized media server in the preferred embodiment ...

Id., 20:11-12; *see also* 10:11-11:8.

317. Vasilevsky describes the computer system having at least one computer-readable medium (*e.g.*, server memory and hard drives). For example, Vasilevsky describes that bookmark database 162 stores bookmarks established in the system, and a memory 118 stores video programming.

When a system user pauses the **playback of a particular program**, he or she can decide whether to "bookmark" the program. That is, an electronic marker can be placed in memory indicating exactly where the program was paused, along with other particulars such as the identity of the user, and the date and time of the bookmark. When a command is received to enter a bookmark, the Application Unit 134 causes the **bookmark** to be **stored in a bookmark database 162**, along with **program identity information** so that the **bookmark** can be matched to the appropriate program **stored in memory 118**.

Id., 19:13-20 (emphasis added).

In the preferred embodiment, the memory 118 is a high-capacity hard disk drive, although other forms of memory systems are compatible. The writing of program content information to memory and the reading of program content information from memory are controlled by a video manager 116.

Id., 14:3-7 (emphasis added).

318. As noted above, Vasilevsky teaches the centralized media server including components 120, 122, 134 (which is the Application Unit), 136 and 152 (which is the Application Services). Vasilevsky's configuration results in processor and memory being at the server-side such that the centralized media server executes software for serving clients such as "'thin' clients" and generating a user interface.

While the components 120, 122, 134, 136 and 152 are part of the centralized media server in the preferred embodiment, it should be understood that these components might be implemented in a client system or other end-user equipment (e.g., 124, 128 and 132), in an alternate embodiment. While today's processor and memory prices dictate that the aforementioned components are better incorporated in the centralized media server to serve "thin" clients, future price reductions (and/or performance increases) in these elements may make the inclusion of these components in client or end-user equipment economically feasible.

Id., 20:11-20 (emphasis added).

The system users communicate with the media server via a user interface or Application Unit 134, which converts user commands to commands compatible with the media server. The **Application Unit 134 is also responsible for generating a graphical user interface (GUI)** containing an electronic program guide (EPG) for display, and input by an Application Services Unit 152.

Id., 15:15-20 (emphasis added).

319. The computer-readable media of Vasilevsky comprises a video editing environment in the form of software for both editing bookmarks and streaming video, executed by the centralized media server and storages. The centralized media server executes software for editing bookmarks and for streaming video.

While the components 120, 122, 134, 136 and 152 are part of the centralized media server in the preferred embodiment, it should be understood that these components might be implemented in a client system or other end-user equipment (e.g., 124, 128 and 132), in an alternate embodiment. While today's processor and memory prices dictate that the aforementioned components are **better incorporated in the centralized media server to serve "thin" clients**, future price reductions (and/or performance increases) in these elements may make the inclusion of these components in client or end-user equipment economically feasible.

Id., 20:11-20 (emphasis added).

A centralized Digital Video Recording (DVR) and reproduction system (100) linking several reproduction and control units (such as television receivers (124, 128, 132), etc.) to a **centralized media server (152) allows system users (124, 128, 132) to** not only **access** the same program or programs from different receivers (106, 110), but to pause and resume the same programs from the same marker points **('bookmarks') established** by any **system user (124, 128, 132).** The program "bookmark" representing the program pause point is conveniently shown via a graphical user interface display with a bookmark icon, including the name of the user (124, 128, 132) who established the bookmark and the exact pause point in the program.

Vasilevsky, Abstract (emphasis added).

When a command is received to enter a bookmark, the **Application Unit 134 causes the bookmark to be stored in a bookmark database 162,** along with program identity information so that the bookmark can be matched to the appropriate program stored in memory 118.

Id., 19:16-20 (emphasis added).

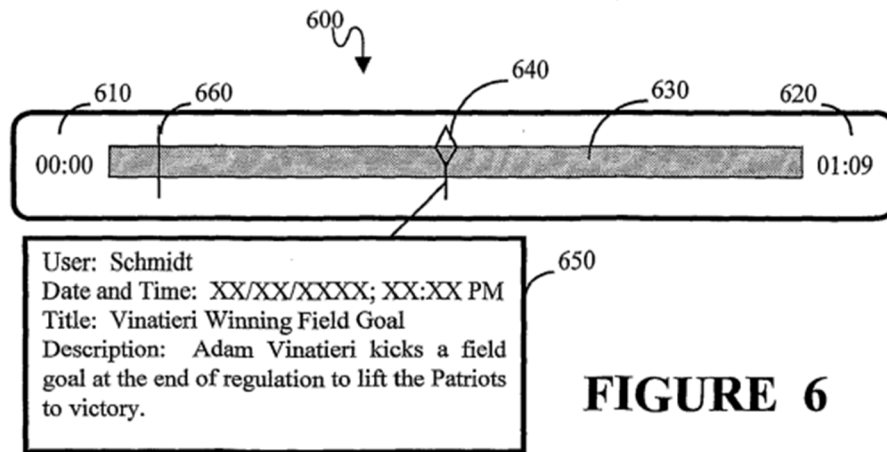


FIGURE 6

Id., FIG. 6.

The system of Claim 1, wherein said **media server is adapted to enable a particular program bookmark to be edited** only by a user purporting to have established the particular bookmark.

Id., claim 5 (emphasis added).

The system of Claim 7, wherein said **bookmark indicia are modifiable by a user**, and said bookmark indicia further comprise bookmark name information.

Id., claim 11 (emphasis added).

The media server is further adapted to allow a user to, from one reproduction device, place a program bookmark in a program representing a point in the program where reproduction has stopped, and from another reproduction device, **resume reproduction of the program from the program bookmark.**

Id., 7:13-17 (emphasis added).

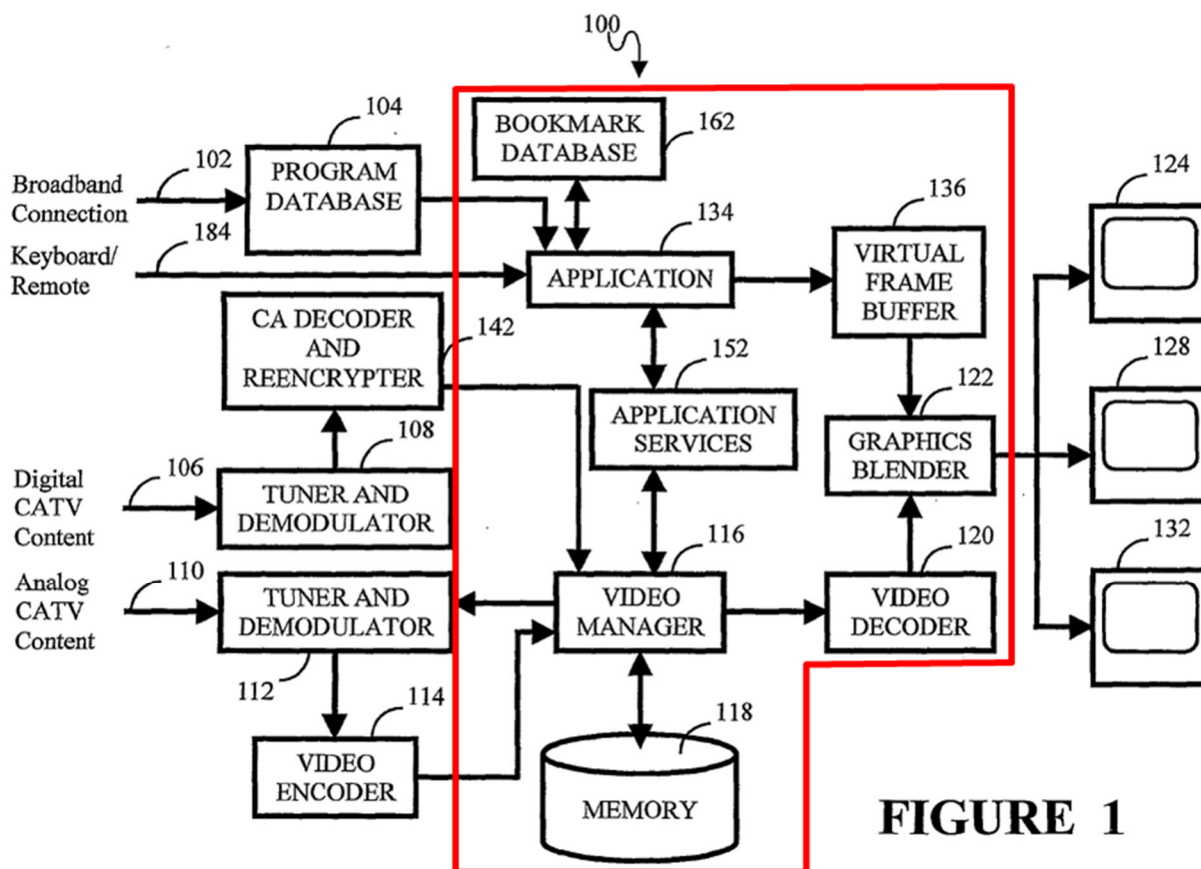


FIGURE 1

Id., FIG. 1 (annotating centralized media server's components).

320. Vasilevsky describes such computer system being disposed at the server side of a public network (*e.g.*, [home area network] HAN or Internet), in contrast to a client side of the HAN (*i.e.*, the side where client 124 is located).

A centralized Digital Video Recording (DVR) and reproduction system (100) **linking** several reproduction and control units (such as television receivers (124, 128, 132), etc.) to a **centralized** media server (152) allows system users (124, 128, 132) to not only access the same program or programs from different receivers (106, 110), but to pause and resume the same programs from the same marker points ('bookmarks') established

by any system user (124, 128, 132). The program "bookmark" representing the program pause point is conveniently shown via a graphical user interface display with a bookmark icon, including the name of the user (124, 128, 132) who established the bookmark and the exact pause point in the program.

Id., Abstract (emphasis added).

The home area network (HAN) subsumes all of the major connections between the components in the system 100, and is essentially a computer bus adapted for both digital communication and the transport of digital video and digital audio multimedia content compatible with digital television receivers.

Id., 11:9-13.

Those skilled in the art will appreciate that the media server can be structured differently from that illustrated, and can include more or fewer of the components in Figure 1. The most important aspect of the **media server** is that it is a **centrally located means** for storing multiple programs that are readily and contemporaneously accessible by, and readily and contemporaneously controlled by multiple local client devices **via a HAN**.

Id., 16:6-12 (emphasis added); *see also* FIG. 1.

321. To the extent Vasilevsky does not explicitly disclose its network (HAN) is a “public” network, Logan discloses a similar bookmarking system to Vasilevsky

and teaches implementing a television program bookmarking system using a public network, *i.e.*, the Internet.

Similarly, **one user could bookmark** an individual program or **a segment of a program, associate a recommendation or comment with the bookmarked content**, and make the program or program segment identification data and the comment or recommendation available to a special interest group or to a specific individual.

Logan, ¶[0223] (emphasis added).

Using the facilities of an **interactive digital cable television networks**, a viewer could be watching a show live and **want to recommend it to another friend**. Using a remote control, the user could select one or more friends from a preset displayed list and then transmit to those designated persons a “watch this” message that might be displayed as close-captioned text on the friend's screen. Properly programmed, the receiver could provide the option to open a window on the TV screen for a PIP (“picture-in-picture”) display of the recommended show. Alternatively, **using the Internet**, a **message could be sent** via an instant bookmarking messaging connection or by email to a designated person or persons.

Id., ¶[0224] (emphasis added).

A remote editing station, which may be at the broadcast facility or at a remote location, classifies, describes or otherwise

identifies individual segments of broadcast programming and sends metadata (sometimes referred to as "markup data") identifying and describing those segments to a remote client receiver. For example, the markup data may identify individual segments by specifying the source and the time of the original broadcast, or by specifying some other unique characteristic of the broadcast signal. The program segments may be TV, radio, or **Internet programs**, or portions of programs, including individual songs, advertisements, or scenes.

Id., ¶[0015] (emphasis added).

The communication link used to **transmit the metadata to the client** may take one of several forms, including the **Internet**, a dialup telephone link, the communications pathway used to carry the broadcast signals to the client, or other forms of communication used to transport the metadata to the client.

Id., ¶[0016] (emphasis added).

The communication methods or apparatus used to transport metadata and/or content to the user as illustrated at 130 may take many different forms, including: the Internet, a dialup telephone connection through the public switched telephone network (PSTN), a wireless transmission system, cable, private line facilities, or data storage media transported from the content publisher and/or the metadata creator to the user. The communications may take place over a combination of such facilities and, as noted earlier, the content and metadata may be

transmitted in one or both directions together or separately over the same or different facilities.

Id., ¶[0050] (emphasis added).

The present invention contemplates the creation and use of metadata for describing and manipulating programming content of the type typically broadcast for public consumption by radio and **television broadcast stations**; disseminated by cable and satellite systems and, more recently, via the **Internet**; or published for general consumption on data storage media, such as DVD disks.

Id., ¶[0062] (emphasis added); *see also* Abstract.

322. A POSITA would have been motivated to modify Vasilevsky's network ("HAN" Home Area Network) to be a public network such as the Internet, as taught by Logan. Logan discloses the desirability of implementing its bookmarking system over a public network such as the Internet, in order to permit members of the public, such as other viewers, to contribute to the system.

The **communication methods or apparatus used to transport metadata and/or content to the user** as illustrated at 130 may take many different forms, including: the **Internet**, a dialup telephone connection through the public switched telephone network (PSTN), a wireless transmission system, cable, private line facilities, or data storage media transported from the content publisher and/or the metadata creator to the user. The

communications may take place over a combination of such facilities and, as noted earlier, the content and metadata may be transmitted in one or both directions together or separately over the same or different facilities.

Logan, ¶[0050] (emphasis added).

In addition, metadata contributed by other users and stored in a **public database 254** as well as private metadata from a database 255 may be downloaded to the PVR by the metadata service's download facility 250. The metadata stored at 251-255 may be created, edited and deleted using a **Web** server 261 or other server 262 operated by the metadata service **to permit the public (other viewers) to contribute to the metadata** as illustrated at 264, as well as providing the ability for employees of the metadata service to create and modify the stored metadata as indicated at 265. Using the PVRs data upload facility seen at 250, usage data, “watch this” selections, playlists, and other user-generated metadata may be uploaded from the PVR via the user data submission facility 272 provided by the metadata service which **may supply** both **the public** and private metadata stores 254 and 255.

Id., ¶[0308] (emphasis added).

323. A POSITA would have been motivated to implement Vasilevsky's system on a public network, as taught by Logan, to allow users to access the system when they are not on their home network (*e.g.*, access could now be from anywhere

in the world as opposed to just over a user's home network). A POSITA would have recognized that such a modification would have beneficially allowed more users access via a wider public network to participate in the system to add, modify, and playback video bookmarks (*e.g.*, this ease of access and expanded access would allow the system to become more popular and allow sharing of bookmarked video clips with users outside the home network).

324. A POSITA would have had a reasonable expectation of success with such a combination. First, Vasilevsky and Logan are both directed to similar systems of adding bookmarks to television programming. *Compare, e.g.*, Vasilevsky, Abstract, 19:12-20:10 *with* Logan, Abstract, ¶¶[0177]-[0178], [0223]-[0224]. Additionally, it would have been a simple and straightforward modification of substituting one network type with another (*i.e.*, substituting a private network with a public network) to obtain predictable results (*e.g.*, swapping out one type of network router/switch technology for another or swapping out one type of network software module for another). A POSITA would have recognized that such a substitution would not have affected the function of Vasilevsky's system (*e.g.*, there would be no change in function from the perspective of a user of the system as the means of communication and connection to the server is "invisible" to a user). It was well-known to a POSITA that public networks and private networks could be swapped for one another. For example:

... As will be understood by people of ordinary skill in the art, networks 120 and 160 can be separate networks or alternatively can be the same network and can be, for instance, a private or public network such as a common Ethernet, internet, intranet computer network system, wireless network or other electronic system.

Devore, ¶[0032] (emphasis added).

325. It would have at least been “obvious to try” a public network because it would have merely involved choosing from a finite number of identified, predictable solutions with a reasonable expectation of success (*i.e.*, the only types of networks to choose from are private or public). Moreover, Vasilevsky’s system is already connected to a public network (WAN)—this teaching would have further motivated such modification (*e.g.*, by simply extending the existing connection to the public network into the HAN).

In the preferred embodiment, the program database 104 obtains programming information via a broadband or wide area network (WAN) connection 102 to a host computer (not shown).

Vasilevsky, 12:8-10 (emphasis added).

326. Finally, a POSITA would have recognized the difficulties and costs associated with creating and maintaining numerous, private in-home servers in each individual home (*e.g.*, the cost of the system is placed on the user who has to install

and maintain the HAN; such cost being multiplied across each home using Vasilevsky's system). A POSITA would have recognized that efficiencies and cost-savings would be gained in placing Vasilevsky's system on a public network (Internet) with a single centralized server (*e.g.*, this would shift the cost to the owner of the Vasilevsky system who would now be responsible to set up and maintain the necessary server(s), and the only cost to the user would be associated with the user would be via their own client device).

3. [13.2] receiving a set of video information at the video editing environment using the computer system;

327. In my opinion, Vasilevsky teaches limitation [13.2].

328. Vasilevsky discloses receiving a set of video information (*e.g.*, receiving a collection of information concerning video recordings) at the video editing environment using the computer system (*e.g.*, at the software for streaming video using the centralized media server and storages).

The present home area-networked digital video recording and playback system 100, capable of program bookmarking and playback resumption of the same program from multiple receivers, is explained below with reference to Figure 1. In its elementary form, the system 100 combines a media server for **receiving and storing multiple electronic audio-visual programs** (*e.g.*, digital or analog television broadcasts, video recordings such as those provided by video-on-demand services, or even audio programs), and several reproduction devices such

as digital or analog television sets (124, 128 and 132) via a home area network (HAN). **The media server acts as a centralized Digital Video Recorder and Playback device for all of the receivers connected to the HAN.** In the preferred embodiment, the media server contains all of the components of the system 100 shown in Figure 1, except the receivers 124, 128 and 132, graphics blender 122, video decoder 120, and the connections 102, 106, 110 and 184. A graphics blender 122 and video decoder 120 can be included in a set-top box for each receiver 124, 128 and 132.

Id., 10:11-11:8 (emphasis added).

The **television signals** and other program signals **are stored in memory 118** in a predefined manner or as dictated by a user using a keyboard or remote control device via input 184.

Id., 13:19-14:1.

329. Vasilevsky explains that the set of video information is received at the video editing environment using the computer system, because recorded videos are, for example, received by software executed by Application Unit 134, Video Manager 116, and Memory 118 when retrieving video for streaming to a client 124.

The **system users communicate with the media server via a user interface or Application Unit 134**, which converts user commands to commands compatible with the media server. The Application Unit 134 is also responsible for generating a graphical user interface (GUI) containing an electronic program

guide (EPG) for display, and input by an Application Services Unit 152. The Application Services Unit 152 controls the recording and playback of programs. Under the direction of the Application Unit 134 and utilizing program guide information, the Application Services Unit 152 establishes the necessary memory needed for recording a particular program, whether the recording is of the traditional or live-pause variety.

Id., 15:15-16:5 (emphasis added).

The writing of program content information to memory and the reading of program content information from memory are **controlled by a video manager 116.**

Id., 14:5-7 (emphasis added).

The **television signals and other program signals are stored in memory 118** in a predefined manner or as dictated by a user using a keyboard or remote control device via input 184. The input signals are in response to a graphical user interface displayed on the receiver screens. In the preferred embodiment, the memory 118 is a high-capacity hard disk drive, although other forms of memory systems are compatible. The writing of program content information to memory and the reading of program content information from memory are controlled by a video manager 116.

Id., 13:19-14:7 (emphasis added).

Figure 2 illustrates an example of a program guide display 200 presented to a system user, which **display contains previously stored programs on the media server**, or those in the process of being viewed in the live-pause mode. By highlighting a particular program, more detailed information can be obtained from the "Program Details" display 300 in Figure 3. The "Program Details" display 300 allows a user to reproduce the particular program from the start, or from a previously established bookmark.

Id., 21:1-8 (emphasis added).

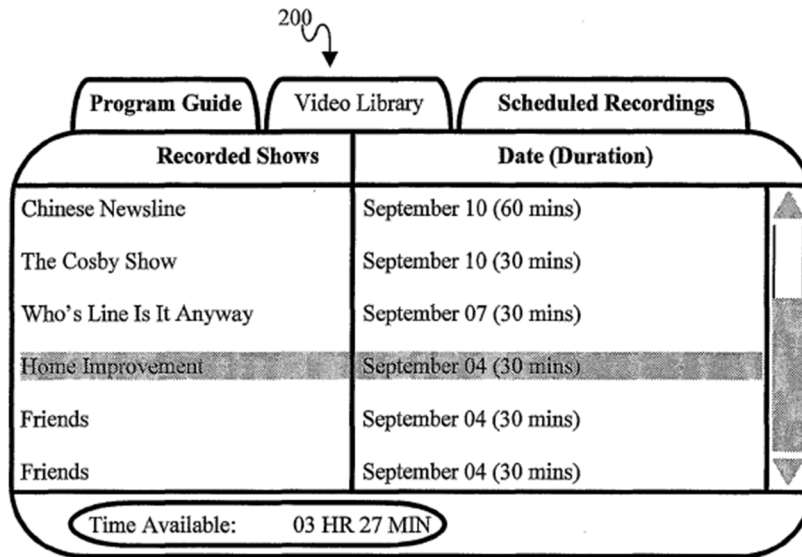


FIGURE 2

Id., FIG. 2 (showing previously stored programs on the centralized media server).

4. [13.3] receiving a set of cue point information at the video editing environment using the computer system;

330. In my opinion, Vasilevsky teaches limitation [13.3].

331. Vasilevsky discloses multiple instances of receiving a set of cue point information (*e.g.*, receiving a collection of information concerning bookmarks) at the video editing environment using the computer system (*e.g.*, at the software for editing bookmarks using the centralized media server and storages) in multiple ways.

332. A first instance of Vasilevski disclosing receiving a set of cue point information is where the media server is configured to allow a client reproduction device to place a bookmark in a program where the program was stopped.

The media server is further adapted to allow a user to, from one **reproduction device, place a program bookmark in a program** representing a **point in the program where reproduction has stopped**, and from another reproduction device, resume reproduction of the program from the program bookmark.

Vasilevsky, 7:14-17 (emphasis added).

... upon the demand of a user via a reproduction device, transmitting in a reproducible format, programs to the reproduction devices, via one reproduction device and the media server, and upon the demand of a user, **placing a program bookmark in a stored program representing a point in the program where reproduction has stopped**, and via another

reproduction device and the media server, and upon the demand of a user, resuming reproduction of the program from the program bookmark.

Id., 8:7-12 (emphasis added).

When a system user pauses the playback of a particular program, he or she can decide whether to "bookmark" the program. That is, an **electronic marker can be placed in memory indicating exactly where the program was paused**, along with other particulars such as the identity of the user, and the date and time of the bookmark.

Id., 19:13-16 (emphasis added).

333. A second instance of Vasilevski disclosing receiving a set of cue point information is where, when a user pauses a program, the user may decide whether to bookmark the pause-point by placing an electronic marker in memory such that the Application Unit 134 creates and stores the bookmark in bookmark database 162.

When a system **user pauses the playback of a particular program**, he or she can **decide whether to "bookmark" the program**. That is, an **electronic marker can be placed in memory** indicating exactly where the program was paused, along with other particulars such as the identity of the user, and the date and time of the bookmark. When a command is received to enter a bookmark, the **Application Unit 134 causes the bookmark to be stored in a bookmark database 162**, along

with program identity information so that the bookmark can be matched to the appropriate program stored in memory 118.

Id., 19:12-20 (emphasis added).

Figure 6 shows a variation 600 of the **bookmark display** 400 in Figure 4. The elements 610, 620, 630, 640, 650 and 660 are analogous to the elements 410, 420, 430, 440, 450 and 460, respectively. However, the identifier 650 is in the form of a more detailed text box that includes such information as the bookmark establisher's name, a title of the bookmark, and a brief description of what occurred in the program just prior to the pause point. **A system user can enter or modify all of the information.**

Id., 22:8-15 (emphasis added).

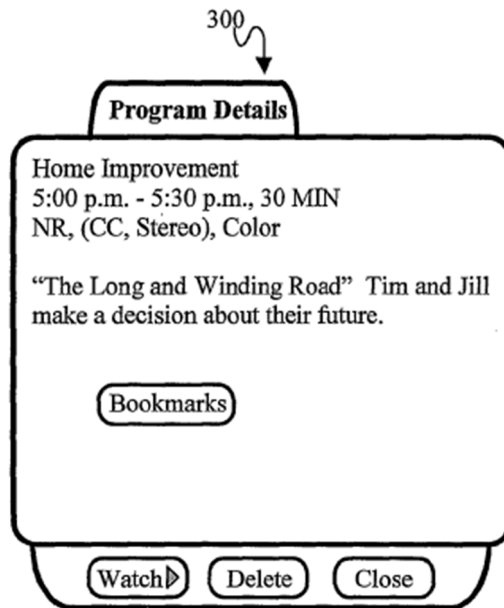


FIGURE 3

Id., FIG. 3.

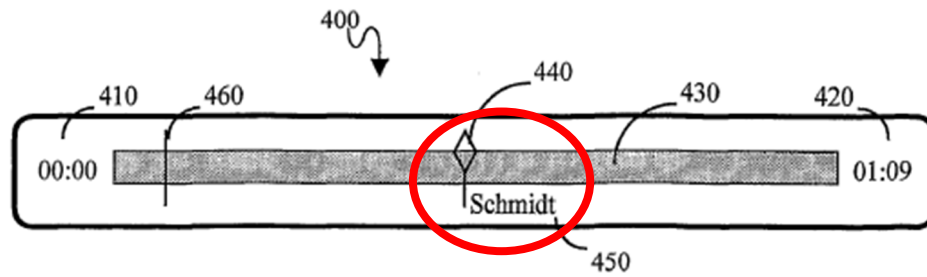


FIGURE 4

Id., FIG. 4 (annotated, showing cue point 440 added by “Schmidt”).

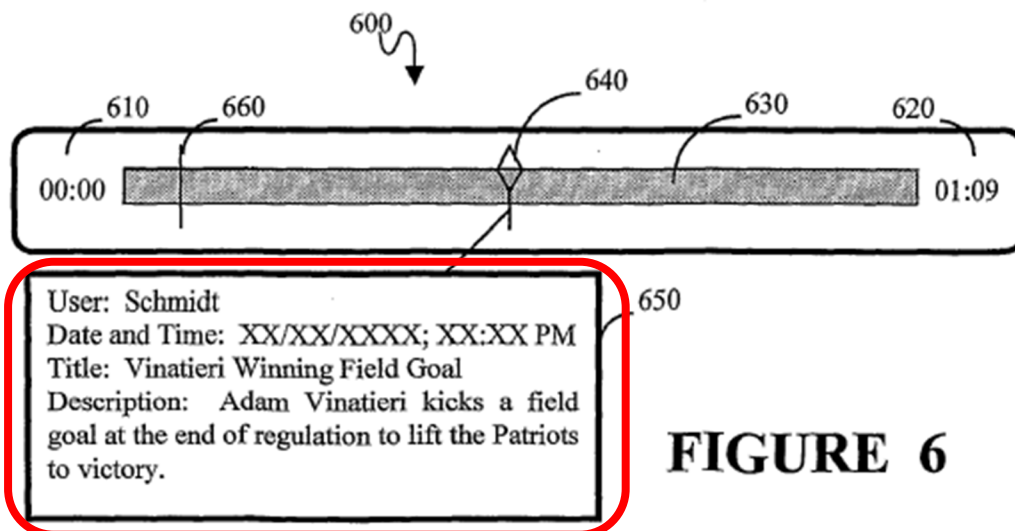


FIGURE 6

Id., FIG. 6 (annotated, showing cue point 640 added by “Schmidt”).

700

BOOKMARK LISTING		
PROGRAM	USER	LOCATION
<u>Home Improvement</u>	<u>Publius Virgil</u>	0 hrs., 13 min., 15 sec. out of 0 hrs., 30 min., 0 sec.
<u>Home Improvement</u>	<u>Mary Shelley</u>	0 hrs., 20 min., 37 sec. out of 0 hrs., 30 min., 0 sec.
<u>NFL: Raiders v. Patriots</u>	<u>Victor Hugo</u>	3 hrs., 3 min., 12 sec. out 3 hrs., 15 min., 0 sec.
<u>XXXXXXXXXXXXXX</u>	<u>XXXXXXXXXXXXXX</u>	<u>XXXXXXXXXXXXXXXXXX</u> <u>XXXXXXXXXXXXXXXXXX</u>
<u>XXXXXXXXXXXXXX</u>	<u>XXXXXXXXXXXXXX</u>	<u>XXXXXXXXXXXXXXXXXX</u> <u>XXXXXXXXXXXXXXXXXX</u>

FIGURE 7

Id., FIG. 7 (annotated, showing cue point information including at least three different bookmarks).

334. Vasilevsky discloses the set of cue point information being received at the video editing environment using the computer system, as bookmarks are received by software executed by the Application Unit 134 and the Bookmark Database 162, including when receiving a new bookmark from the client 124 or retrieving a bookmark that the client 124 is to edit.

The system users communicate with the media server via a user interface or Application Unit 134, which converts user commands to commands compatible with the media server. The Application Unit 134 is also responsible for generating a graphical user interface (GUI) containing an electronic program guide (EPG) for display, and input by an Application Services Unit 152.

Id., 15:15-20.

When a system user pauses the playback of a particular program, he or she can decide whether to "bookmark" the program. That is, an electronic marker can be placed in memory indicating exactly where the program was paused, along with other particulars such as the identity of the user, and the date and time of the bookmark. When a command is received to enter a bookmark, the Application Unit 134 causes the bookmark to be stored in a bookmark database 162, along with program identity information so that the bookmark can be matched to the appropriate program stored in memory 118.

Id., 19:12-20; *see also* 21:1-8.

5. **[13.4] receiving a cue point editing command over the network from a remote client terminal disposed at a client side of the public network using the computer system;**

335. In my opinion, Vasilevsky teaches limitation [13.4].

336. Vasilevsky discloses receiving a cue point editing command (*e.g.*, command to edit a bookmark after the client 124 modifies bookmark information in FIG. 6 interface) over the network (*e.g.*, HAN or Internet) from a remote client terminal disposed at a client side of the public network (*e.g.*, client 124) using the computer system (*e.g.*, the centralized media server and storages).

337. Vasilevsky's FIG. 6 illustrates a bookmark text box that includes information as the bookmark establisher's name, a title of the bookmark, and a brief description. To edit a bookmark, the client 124 enters an edit mode by selecting a bookmark and has the ability to "modify all of the information" resulting in a command to edit the bookmark being sent from client 124 to the centralized media server.

Figure 6 shows a variation 600 of the bookmark display 400 in Figure 4. The elements 610, 620, 630, 640, 650 and 660 are analogous to the elements 410, 420, 430, 440, 450 and 460, respectively. However, the identifier 650 is in the form of a more detailed **text box** that includes such information as the bookmark establisher's **name**, a **title** of the bookmark, and a **brief description** of what occurred in the program just prior to the pause point. **A system user can** enter or **modify all of the information**.

Vasilevsky, 22:8-15 (emphasis added).

The system of Claim 1, wherein said media server is adapted to enable any **program bookmark** to be **edited** by any **authorized system user**.

Id., claim 4 (emphasis added).

The system of Claim 7, wherein said bookmark indicia are **modifiable by a user**, and said bookmark indicia further comprise bookmark **name** information.

Id., claim 11 (emphasis added).

338. The computer system receives the command over the network (*e.g.*, HAN or Internet).

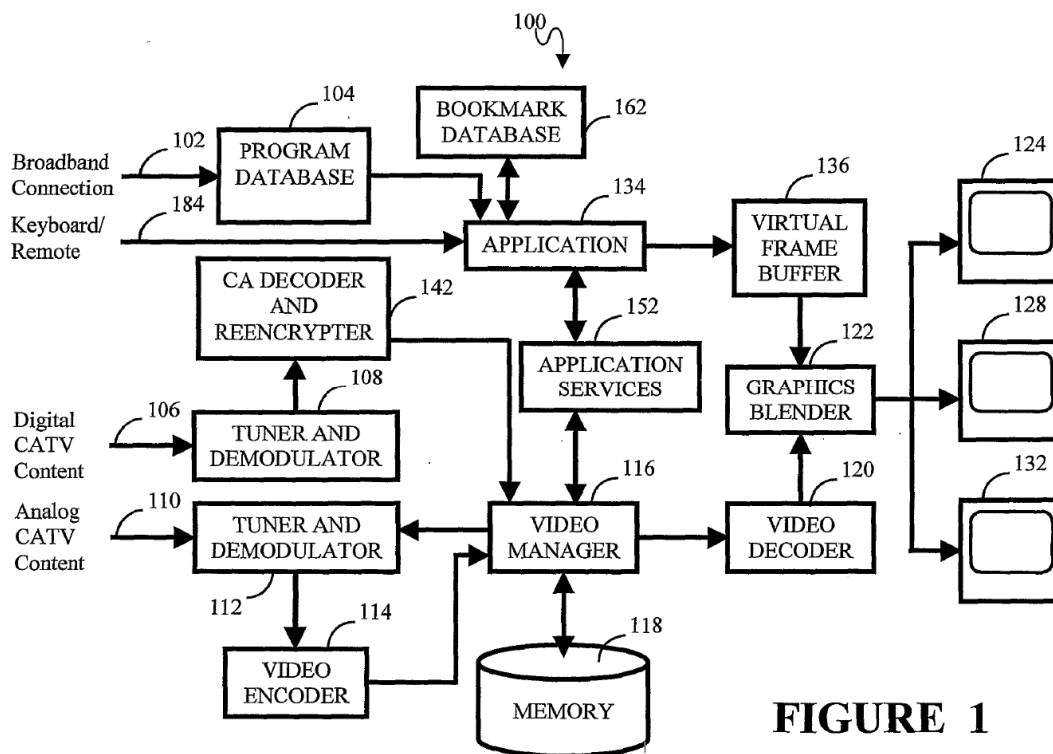


FIGURE 1

Id., FIG. 1.

The system users communicate with the media server via a user interface or **Application Unit 134**, which **converts user commands to commands compatible with the media server**.

The Application Unit 134 is also responsible for generating a graphical user interface (GUI) containing an electronic program guide (EPG) for display, and input by an Application Services Unit 152.

Id., 15:15-20 (emphasis added).

When a system user pauses the playback of a particular program, he or she can decide whether to "bookmark" the program. That is, an electronic marker can be placed in memory indicating exactly where the program was paused, along with other particulars such as the identity of the user, and the date and time of the bookmark. **When a command is received to enter a bookmark, the Application Unit 134 causes the bookmark to be stored in a bookmark database 162**, along with program identity information so that the bookmark can be matched to the appropriate program stored in memory 118.

Id., 19:12-20 (emphasis added).

339. A POSITA would have understood the plain meaning of "cue point editing command" to be a command, sent from the remote client terminal and received by the computer system, to edit a portion of the cue point information. *See* limitation [13.5] (indicating that "according to **the** cue point editing command" "a portion of the cue point information" is "modif[ied]") (emphasis added). Further, a

POSITA would have understood the word “command” (in “cue point editing command”) to mean an instruction to a computer program, due to how this word is used in the ’286 Patent.

340. The ’286 Patent describes that “the **command**, ‘RETRIEVE (‘VID—0001’), is used to retrieve a video file identified as ‘VID—0001’ from the video server 112. When the video server 112 receives the ‘RETRIEVE’ **command**, a relational database may be queried to find an entry for ‘VID—0001’ ... The video server 112 may then use various algorithms to select a download location and download the file.” ’286 Patent, 7:33-41 (emphasis added). In other words, an instruction is sent to the video server to retrieve a video, to which the video server responds. *See also id.*, 13:65-67 (describing “Visual Basic **commands** for an interpreter” that are instructions to the interpreter “to generate the GUI”) (emphasis added), 42:45-48 (describing a “**command** 1640” “transmitted by a client, received by a host or application” “requesting that a dataset is built for all ‘Exterior’ cue points” and the host or application responds to the instruction by “generat[ing] dataset 1650.”) (emphasis added).

341. This understanding, that a “command” is an instruction to a computer program, also comports with contemporaneous technical dictionaries, *e.g.*, the Microsoft Computer Dictionary defines a “command” as “An instruction to a

computer program that, when issued by the user, causes an action to be carried out.”
Exhibit 1017.

342. Therefore, a POSITA would have understood that the user’s changing bookmark information in the FIG. 6 interface results in sending a cue point editing command—an instruction to the computer system to edit a portion of the annotation—because the computer system responds by carrying out the requested action (*see* limitation [13.5] below).

343. Vasilevsky’s client 124 is disposed at a client side of the network, in contrast to a server side of the network where the centralized media server and storages are located and is therefore also remote from the servers.

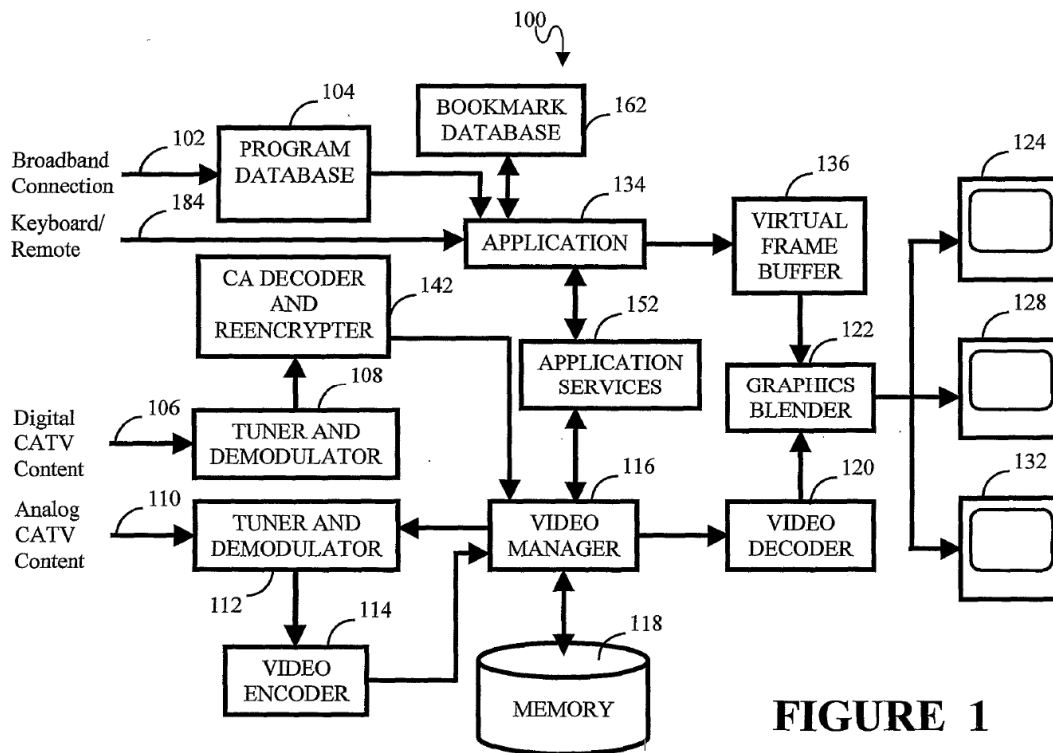


FIGURE 1

Id., FIG. 1.

The media server acts as a **centralized** Digital Video Recorder and Playback device for all of the receivers **connected to the HAN**.

Id., 11:1-2 (emphasis added); *see also id.*, claims 4-5.

6. [13.5] modifying a portion of the cue point information at the server side of the public network according to the cue point editing command received over the public network from the client terminal;

344. In my opinion, Vasilevsky teaches limitation [13.5].

345. Vasilevsky discloses modifying a portion of the cue point information (*e.g.*, editing bookmark information) at the server side of the public network (*e.g.*, at the centralized media server) according to the cue point editing command received over the public network from the client terminal (*e.g.*, the command received in limitation [13.4]).

346. Vasilevski discloses that when the client 124 modifies existing bookmarks in the FIG. 6 interface, the centralized media server modifies the bookmarks and the Application Unit 134 stores the modified bookmarks in bookmark database 162. *See* my discussion above of limitation [13.4]; *see also*:

When a system user pauses the playback of a particular program, he or she can decide whether to "bookmark" the program. That is, an electronic marker can be placed in memory indicating exactly where the program was paused, along with other

particulars such as the identity of the user, and the date and time of the bookmark. When a command is received to enter a bookmark, the Application Unit 134 causes the bookmark to be stored in a bookmark database 162, along with program identity information so that the bookmark can be matched to the appropriate program stored in memory 118.

Vasilevsky, 19:12-20.

Figure 6 shows a variation 600 of the bookmark display 400 in Figure 4. The elements 610, 620, 630, 640, 650 and 660 are analogous to the elements 410, 420, 430, 440, 450 and 460, respectively. However, the identifier 650 is in the form of a more detailed text box that includes such information as the bookmark establisher's name, a title of the bookmark, and a brief description of what occurred in the program just prior to the pause point. A system user can enter or modify all of the information.

Id., 22:8-15; *see also id.*, 21:9-20, 15:15-17, claims 4-5.

7. **[13.6] receiving a cue point selection over the network from the remote client terminal identifying one of the cue points corresponding to a portion of the video information of one of the video files; and**

347. In my opinion, Vasilevsky teaches limitation [13.6].

348. Vasilevsky discloses receiving a cue point selection (*e.g.*, receiving a selected bookmark based on a request to resume a program based on a selected bookmark) over the network from the remote client terminal (*e.g.*, from client 124

over HAN or Internet) identifying one of the cue points corresponding to a portion of the video information of one of the video files (*e.g.*, selected bookmark identifies a video portion).

It is therefore desirable to significantly improve the prior art by providing a system with a seamless pause and playback feature where either pause or **resumption of playback from an established "bookmark" can occur from any of the receivers,** and without the increased cost of related prior art approaches.

Vasilevsky, 6:11-15 (emphasis added).

... allow a user to, from one reproduction device, place a program bookmark in a program representing a point in the program where reproduction has stopped, and from another reproduction device, **resume reproduction of the program from the program bookmark.**

Id., 7:14-17 (emphasis added).

Alternatively, or in conjunction, **clicking on the bookmark** can lead directly to the pause point in the program of the representative bookmark.

Id., 23:2-24 (emphasis added); *see also id.*, 8:7-12, 18:13-14, 18:19-19:2, 20:3-6.

349. Additionally, Vasilevsky discloses the cue point selection being received by the computer system over the network (*e.g.*, HAN or Internet) from

client 124 as the centralized media server and the client reproduction devices are interconnected via the HAN or Internet.

In its elementary form, the system 100 combines a media server for receiving and storing multiple electronic audio-visual programs (e.g., digital or analog television broadcasts, video recordings such as those provided by video-on-demand services, or even audio programs), and several reproduction devices such as digital or analog television sets (124, 128 and 132) via a home area network (HAN). The **media server** acts as a **centralized** Digital Video Recorder and Playback device for all of the **receivers connected to the HAN**.

Id., 10:14-11:2 (emphasis added).

The system users communicate with the media server via a user interface or Application Unit 134, which converts user commands to commands compatible with the media server.

Id., 15:15-17.

350. Finally, Vasilevsky discloses each bookmark (cue point) identifying a portion (video segment with a start and end time) of a video. *See* my discussion above of limitation [13.3]; *see also*:

The bookmark display 400 contains a start time display area 410 and end time display area 420, indicating the program length.

Vasilevsky, 21:11-13 (emphasis added).

When a system user pauses the playback of a particular program, he or she can decide whether to ‘bookmark’ the program. That is, **an electronic marker** can be placed in memory **indicating exactly where the program was paused** ...

Id., 19:12-16 (emphasis added).

Figure 6 shows a variation 600 of the bookmark display 400 in Figure 4. The elements 610, 620, 630, 640, 650 and 660 are analogous to the elements 410, 420, 430, 440, 450 and 460, respectively. However, the identifier 650 is in the form of a more detailed text box that includes such information as the bookmark establisher’s name, a title of the bookmark, and a brief description of what occurred in the program just prior to the pause point. A system user can enter or modify all of the information.

Id., 22:8-15; *see also* FIGs. 4, 6.

8. [13.7] streaming the portion of the video information to the client terminal over the public network in response to receiving the cue point selection.

351. In my opinion, Vasilevsky teaches limitation [13.7].

352. Vasilevsky discloses streaming the portion of the video information to the client terminal (*e.g.*, streaming the portion of the video corresponding to the selected bookmark to client 124) over the public network (*e.g.*, HAN or Internet) in response to receiving the cue point selection (*e.g.*, selected bookmark in the request to resume program from bookmark received in limitation [13.6]).

353. Vasilevsky discloses that when the client 124 selects a bookmark, the centralized media server streams the program from the start point of the bookmark to the client.

The communication and transport of program information between the media server and the various user control and reproduction devices (or client devices, or receivers) can be via a number of network and streaming protocols, including, inter alia: Real-Time Transport Protocol (RTP); Real-Time Streaming Protocol (RTSP); Transmission Control Protocol (TCP); User Datagram Protocol (UDP); Network File System (NFS) Protocol; Web-Distributed Authoring and Versioning (WebDAV) Protocol; Server Message Block (SMB) Protocol; IEEE 1394 Protocol; and Internet Small Computer System Interface (iSCSI) Protocol.

Vasilevsky, 15:5-14 (emphasis added).

The media server is further adapted to allow a user to, from one reproduction device, place a program bookmark in a program representing a point in the program where reproduction has stopped, and from another reproduction device, resume reproduction of the program from the program bookmark.

Id., 7:13-17 (emphasis added); *see also id.*, 21:11-13, Abstract.

D. Claim 14

354. Claim 14 recites:

14. The method of claim 13, further comprising providing a cue point editing mode as part of the video editing environment configured to perform at least one of creating, modifying, deleting, or interpreting at least a portion of the set of cue point information according to the cue point editing command received over the public network from the client terminal.

355. In my opinion, Claim 14 is obvious over Vasilevsky and Logan.

356. Vasilevsky discloses the limitations recited in Claim 14, as Vasilevsky discloses providing a cue point editing mode as part of the video editing environment (*e.g.*, providing interface(s) for editing bookmarks, as part of the software executed by the centralized media unit and storages) configured to perform at least one of creating, modifying, deleting, or interpreting at least a portion of the set of cue point information (*e.g.*, editing bookmark information) according to the cue point editing command received over the public network from the client terminal (*e.g.*, command received in limitation [13.4]).

357. Vasilevsky discloses that to edit an existing bookmark, client 124 enters a cue point edit mode by selecting an existing bookmark so that it is displayed in the FIG. 6 interface where “information [such] as the bookmark establisher’s name, a title of the bookmark, and a brief description of what occurred in the program just prior to the pause point” is displayed and “[a] system user can enter or modify all of the information.”

Figure 6 shows a variation 600 of the bookmark display 400 in Figure 4. The elements 610, 620, 630, 640, 650 and 660 are analogous to the elements 410, 420, 430, 440, 450 and 460, respectively. However, the identifier 650 is in the form of a more detailed text box that includes such information as the bookmark establisher's name, a title of the bookmark, and a brief description of what occurred in the program just prior to the pause point. A system user can enter or modify all of the information.

Vasilevsky, 22:8-15 (emphasis added), *see also id.*, claims 5, 11.

358. Vasilevsky discloses that the centralized media server modifies at least a portion of the bookmark information according to the editing command received over the network from client 124, and Application Unit 134 stores the modified bookmarks in the bookmark database 162. *See my discussion above of limitations [13.4]-[13.5]; see also:*

Application Unit 134, which converts user commands to commands compatible with the media server ...

Vasilevsky, 15:15-17.

... when a command is received to enter a bookmark, the Application Unit 134 causes the bookmark to be stored in a bookmark database 162.

Id., 19:16-20.

E. Claim 15

359. Claim 15 recites:

15. The method of claim 14, further comprising: receiving an edit mode entry signal over the network from the client terminal representing entry into the cue point edit mode, and switching the video editing environment to the cue point editing mode in response to receiving the edit mode entry signal.

360. In my opinion, Claim 15 is obvious over Vasilevsky and Logan.

361. Vasilevsky discloses the limitations recited in Claim 15, as Vasilevsky discloses receiving an edit mode entry signal (*e.g.*, a signal for entry to the interface(s) to edit a bookmark, from selecting the interface of FIG. 6) over the network from the client terminal (*e.g.*, from client 124 over the HAN or Internet) representing entry into the cue point edit mode (*e.g.*, representing entry into the interface(s) to edit a bookmark), and switching the video editing environment to the cue point editing mode in response to receiving the edit mode entry signal (*e.g.*, switching the software executed by the centralized media server and storages for displaying the interface(s) for editing the bookmarks, including the editable bookmark fields in the interface of FIG. 6). *See* my discussion above of limitation [13.4] and claim 14; *see also*:

Figure 6 shows a variation 600 of the bookmark display 400

in Figure 4. The elements 610, 620, 630, 640, 650 and 660 are analogous to the elements 410, 420, 430, 440, 450 and 460,

respectively. However, the identifier 650 is in the form of a more detailed text box that includes such information as the bookmark establisher's name, a title of the bookmark, and a brief description of what occurred in the program just prior to the pause point. A system user can enter or modify all of the information.

Vasilevsky, 22:8-15 (emphasis added); *see also* 19:16-20.

362. A POSITA would have understood the plain meaning of “edit mode entry signal” to be a signal, sent from the remote client terminal and received by the computer system, to enter an edit mode. Further, a POSITA would have understood the word “signal” (in “edit mode entry signal”) to mean an indication. *See* Exhibit 1018 (“Signal.” Merriam-Webster.com Dictionary, Merriam-Webster, <https://www.merriam-webster.com/dictionary/signal> (accessed March 29, 2024) (defining “signal” as “1: SIGN, INDICATION”)).

363. As the '286 Patent explains, “in some embodiments, cue point edit mode will always be available, while other embodiments may require active entry into that mode, for example by polling for input.” '286 Patent, 15:28-31 (emphasis added). The “input” may be “a particular event from an input device, like a mouse or keyboard.” *Id.*, 15:31-33; *see also id.*, 18:42:49 (explaining “the client may enter the cue point edit mode in any useful way” including “click[ing] on a

video image” or “select[ing] a GUI control dedicated to providing entry into the cue point edit mode” or “select[ing] a portion of the video progress bar”).

364. Therefore, a POSITA would have understood that when a user selects the interface of FIG. 6 for a selected annotation, the centralized media server receives an edit mode entry signal—an indication to the computer system to enter an edit mode.

F. Claim 18

365. Claim 18 recites:

18. The method of claim 13, further comprising receiving, over the network, authentication information relating to access authority of the remote client terminal.

366. In my opinion, Claim 18 is obvious over Vasilevsky and Logan.

367. Vasilevsky discloses the limitations recited in Claim 18, as Vasilevsky discloses receiving, over the network, authentication information (*e.g.*, information concerning the user’s identity) relating to access authority of the remote client terminal (*e.g.*, indicating who may access or edit a bookmark). The authentication information is received over the network, since such information is passed from the user to the centralized media server.

In the preferred embodiment, any user with access to the system can place or establish program bookmarks in a program being reproduced. Any user can also access the bookmarks of other users and resume the playback of any available program from

any bookmark, regardless of who established it. Furthermore, bookmarking and resumption of playback can be carried out from any system receiver. In an alternate embodiment, **only a user who established** a particular bookmark **can use that particular bookmark** for resumption of playback. Using this latter approach, it may be desirable to display to particular users, only those bookmarks that they have in fact established.

Vasilevsky, 20:1-10 (emphasis added).

Activating a "bookmarks" button in the display 200 causes the GUI to produce a toolbar or the like, which shows the bookmarks that have been established for the program. The bookmark display 400 contains a start time display area 410 and end time display area 420, indicating the program length. A program length bar 430 is shown in the middle of the display. In the example shown, a bookmark icon 440 represents a program bookmark that has been established by User Schmidt (identifier 450). Position 460 can represent a previous bookmark of a user that has been updated to a new position. **The bookmark display 400 can contain** all of the active bookmarks for the program, or **only those to which the user has access.**

Id., 21:9-20 (emphasis added).

5. The system of Claim 1, wherein said media server is adapted to enable a particular program bookmark to be **edited only by a user purporting to have established the particular bookmark.**

Id., claim 5 (emphasis added).

6. The system of Claim 1, wherein said media server is adapted to enable access to a particular bookmark and program resumption from that bookmark **only if the user requesting playback of the program purports to have established the particular bookmark**, and said media server is further adapted to enable general reproduction of a program from a reproduction device regardless of whether the user controlling the reproduction device purports to have established the particular bookmark.

Id., claim 6 (emphasis added).

368. In order for Vasilevsky's system to permit only a user who has access to perform specific interactions with bookmarks, such as edit and/or view bookmarks, Vasilevsky's system incorporates the media server receiving authentication information from the client device to permit identifying the user in question, so the media server has the necessary information to determine whether this particular user has sufficient access for a particular bookmark.

G. Claim 19

369. Claim 19 recites:

19. The method of claim 18, further comprising denying or allowing access to all or part of the video editing environment based at least in part on the access authority of the remote client terminal.

370. In my opinion, Claim 19 is obvious over Vasilevsky and Logan.

371. Vasilevsky discloses the limitations recited in Claim 19, as Vasilevsky discloses denying or allowing access to all or part of the video editing environment (*e.g.*, denying client 124 access to part of the software executed by the centralized media server and storages; denying access to certain bookmarks) based at least in part on the access authority of the remote client terminal (*e.g.*, only the user who created the bookmark can access or edit it).

In the preferred embodiment, any user with access to the system can place or establish program bookmarks in a program being reproduced. Any user can also access the bookmarks of other users and resume the playback of any available program from any bookmark, regardless of who established it. Furthermore, bookmarking and resumption of playback can be carried out from any system receiver. In an alternate embodiment, **only a user who established** a particular bookmark can use that particular bookmark for resumption of playback. Using this latter approach, it may be desirable to display to particular users, only those bookmarks that they have in fact established.

Vasilevsky, 20:1-10 (emphasis added).

Activating a "bookmarks" button in the display 200 causes the GUI to produce a toolbar or the like, which shows the bookmarks that have been established for the program. The bookmark display 400 contains a start time display area 410 and end time

display area 420, indicating the program length. A program length bar 430 is shown in the middle of the display. In the example shown, a bookmark icon 440 represents a program bookmark that has been established by User Schmidt (identifier 450). Position 460 can represent a previous bookmark of a user that has been updated to a new position. **The bookmark display 400 can contain all of the active bookmarks for the program, or only those to which the user has access.**

Id., 21:9-20 (emphasis added).

5. The system of Claim 1, wherein said media server is adapted to enable a particular program bookmark to be **edited only by a user purporting to have established the particular bookmark.**

Id., claim 5 (emphasis added).

6. The system of Claim 1, wherein said media server is adapted to enable access to a particular bookmark and program resumption from that bookmark **only if the user requesting playback of the program purports to have established the particular bookmark,** and said media server is further adapted to enable general reproduction of a program from a reproduction device regardless of whether the user controlling the reproduction device purports to have established the particular bookmark.

Id., claim 6 (emphasis added).

18. The method of Claim 15, further comprising the step of: via said media server, enabling any program bookmark to be edited by any **authorized** system user.

Id., claim 18 (emphasis added).

H. Claim 20

372. Claim 20 recites:

The method of claim 13, wherein the server video data is streamed progressively to a video playback environment at the remote client terminal, allowing the video player to begin displaying streamed server video data while remaining server video data continues to be streamed.

373. In my opinion, Claim 20 is obvious over Vasilevsky and Logan.

374. Vasilevsky discloses the limitations recited in Claim 20, as Vasilevsky discloses server video data is streamed progressively to a video playback environment (*e.g.*, by streaming video recordings progressively to a client display) at the remote client terminal (*e.g.*, client 124), allowing the video player on the client to begin displaying streamed server video data while remaining server video data continues to be streamed (*e.g.*, client 124 displays the streamed video data as it is received). *See* my discussion above of limitation [13.7]; *see also*:

The communication and transport of program information between the media server and the various user control and reproduction devices (or client devices, or receivers) can be via

a number of **network and streaming protocols**, including, inter alia: **Real-Time Transport Protocol (RTP)**; **Real-Time Streaming Protocol (RTSP)**; Transmission Control Protocol (TCP); User Datagram Protocol (UDP); Network File System (NFS) Protocol; Web-Distributed Authoring and Versioning (WebDAV) Protocol; Server Message Block (SMB) Protocol; IEEE 1394 Protocol; and Internet Small Computer System Interface (iSCSI) Protocol.

Vasilevsky, 15:5-14.

The most important aspect of the media server is that it is a centrally located means for storing multiple programs that are readily and contemporaneously accessible by, and readily and contemporaneously controlled by multiple local client devices via a HAN.

Id., 16:8-12.

A graphics blender 122 performs the function of blending disparate video streams into one signal compatible with the displays 124, 128, and 132.

Id., 18:4-5.

A POSITA would have understood the word “progressively” (in “streamed progressively”) to mean streaming that allows display of the streamed server video data to begin while the remaining server video data is still being streamed, based on both the claim language and the ’286 Patent. *See* claim 20 (“streaming progressively

... allowing the video player to begin displaying streamed server video data while remaining server video data continues to be streamed.”); ’286 Patent 25:41-47 (“Further, video files may load in various ways, including as a complete file, streaming, or progressive. Progressive loading of video files, for example, allows the video frame to begin playing the portion of video data which has already loaded while the remainder of video data continues to load.”), 4:7-10 (“client video data is transmitted progressively, allowing the video player to begin displaying transmitted client video data while remaining client video data continues to be transmitted.”).

XV. CONCLUSION

375. For the foregoing reasons, it is my opinion that claims 13-15 and 18-20 of the ’286 Patent recite subject matter that is anticipated and/or would have been obvious to a POSITA.

376. All statements made herein of my own knowledge are true, and all statements made on information and belief are believed to be true. Further, I am aware that these statements are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001, and may jeopardize the validity of the application or any patent issuing thereon. I declare under penalty of perjury that the foregoing is true and correct.

Henry H. Houh.

Henry H. Houh, Ph.D.

Executed on March 29, 2024