

Invalidity Contentions for U.S. Patent No. 6,842,796
Based on U.S. Patent No. 6,311,194 to Amit Sheth et al. (“Sheth”)

Based upon Plaintiff’s Complaint, Infringement Contentions, and apparent claim constructions and application of the claims to Defendant’s accused products, as best as they can be deciphered, the reference charted below anticipates or at least renders obvious the asserted claims. These invalidity contentions are not an admission by Defendant that the accused products are covered by or infringe the asserted claims, particularly when these claims are properly construed and applied. These invalidity contentions are not an admission that Defendant concedes or acquiesces to any claim construction implied or suggested by Plaintiff’s Complaint or Infringement Contentions. Nor is Defendant asserting any claim construction positions through these charts, including whether the preamble is a limitation. The portions of the prior art reference cited below are not exhaustive but are exemplary in nature.

U.S. Patent No. 6,311,194 to Amit Sheth et al. (“Sheth”) was filed on August 21, 2000 and issued on October 30, 2001. Accordingly, Sheth is prior art under at least 35 U.S.C. § 102(e), and 103(a). As described in the following claim chart, the asserted claims of U.S. Patent No. 6,842,796 (the “’796 Patent”), are invalid as anticipated by Sheth.

To the extent that Sheth is found not to anticipate one or more of the asserted claims of the ’796 Patent, these claims are invalid as obvious in view of Sheth, alone or in combination with other prior art references disclosed in Defendant’s Invalidity Contentions and accompanying charts, including without limitation as set forth below.

Additionally, Defendant hereby incorporates by reference Ground 1 presented in *Palo Alto Networks, Inc. v. Taasera Licensing LLC*, IPR2023-00443.

Defendant reserves the right to amend or supplement this claim chart at a later date as more fully set forth in the Cover Pleading.

‘796 Patent	<u>Sheth</u>
[1P] A method of automatically processing an input sequence of data symbols, the method comprising the steps of:	To the extent the preamble of claim 1 is found to be limiting, <u>Sheth</u> it explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation. <i>See, e.g., See, e.g.,</i> Abstract

'796 Patent	<u>Sheth</u>
	A system and method for creating a database of metadata (metabase) of a variety of digital media content, including TV and radio content delivered on Internet. This semantic-based method captures and enhances domain or subject specific metadata of digital media content, including the specific meaning and intended use of original content. To support semantics, a WorldModel is provided that includes specific domain knowledge, ontologies as well as a set of rules relevant to the original content. The metabase may also be dynamic in that it may track changes to the any variety of accessible content, including live and archival TV and radio programming. 4:52-5:55: SUMMARY OF THE INVENTION The present invention is directed to software, a system and a method for creating a database of metadata (metabase) of a variety of digital media content, data sets, including TV and radio content potentially delivered on Internet. The data sets may be accessed locally or remotely via a suitable communications channel such as the Internet. This semantic-based method captures and enhances domain or subject specific metadata of digital media content, including the specific meaning and intended use of original content. The digital media content can be semi-structured text, audio, video, animations, etc. To support semantics, the present invention uses a WorldModel that includes specific domain knowledge, ontologies as well as a set of rules relevant to the original content. The metabase is also dynamic because it may track changes to locally or remotely accessible content, including live and archival TV and radio programming. Because these tasks would be labor intensive if performed manually, two methods and apparatus have been designed and implemented. First, a distributed method and apparatus to quickly produce agents which automatically create and manage digital media metadata. Second, a WorldModel that embodies the essence of semantics that is used by the agents and captured in the metadata they produce. The WorldModel cooperates with an associated Knowledge base that uses semantics to enhance relevant information that may not be present in the original source. Assets, profile and personalization

'796 Patent	<u>Sheth</u>
	information as well as advertisement and e-commerce are correlated through the WorldModel. The metabase created by this system represents a unique and proprietary map of a part of the Web, and achieves one unique form of the realization of a Semantic Web. The WorldModel forms the basis of implementing a Semantic Web as well as for developing methods to support applications of Semantic Web, including Semantic Search, Semantic Profiling and Semantic Advertisement. Semantic search is the first application of a Semantic Web and consists of a set of methods for browsing and searching. Accordingly, a variety of interfaces and look-and-feel methods have been developed. These interface methods and apparatus support semantics and context with respect to information content domain and/or original content perspective of radio, television, the Web, etc. Additional methods have been developed to utilize semantics related to user's current interest and information need, as well as historical information, to achieve a semantics-based personalization and profiling (Semantic Profiling) and semantics-based targeted advertisements (Semantic Advertisement). Semantics may be exchanged and utilized between partners, including content owner (or content syndicator or distributor), destination sites (or the sites visited by users), and advertisers (or advertisement distributors or syndicators), to improve the value of content ownership, advertisement space (impressions), and advertisement charges. A one-click-media-play option is provided that utilizes content rights and ownership information to enhance user experience as well as bring increased revenues through better customization and targeting of advertisement and e-commerce for partners involved in Semantic Web applications. 17:38-65: In summary, the present invention is directed to a method for generating metadata associated with a data set residing on a storage system. The method comprises steps of: receiving a classifier associated with a data set; retrieving WorldModel-based extraction information associated with the received classifier; extracting metadata

'796 Patent	<u>Sheth</u>
	associated with the data set based upon the retrieved extraction information; and storing in a data store the extracted metadata which is correlated with an identifier corresponding to the data set. Moreover, the present invention is directed to a method for generating targeted content, comprising steps of receiving a classifier associated with a data set; retrieving WorldModel-based extraction information associated with the received classifier; extracting metadata associated with the data set based upon the retrieved extraction information; and accessing content information based on the metadata for presentation to a user. In addition, the present invention is directed to a system for generating metadata associated with a data set, the system comprising: a processor, and a data store coupled to the processor and storing WorldModel-based extraction information. The processor is programmed to retrieve a classifier associated with a data set; retrieve WorldModel-based extraction information associated with a received classifier; extract metadata associated with the data set based upon the retrieved extraction information; and storing in the data store the extracted metadata which is correlated with an identifier corresponding to the data set. FIG. 5, 6:6-7, 10:7-67, 11:11-22, 11:37-55, 12:34-13:2:

'796 Patent

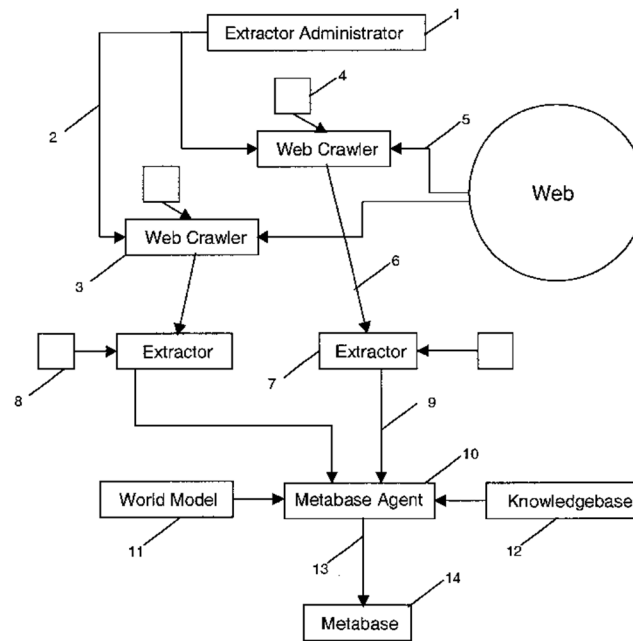
Sheth**Fig. 5**

FIG. 5 is a block diagram of a software architecture according to the present invention.

...

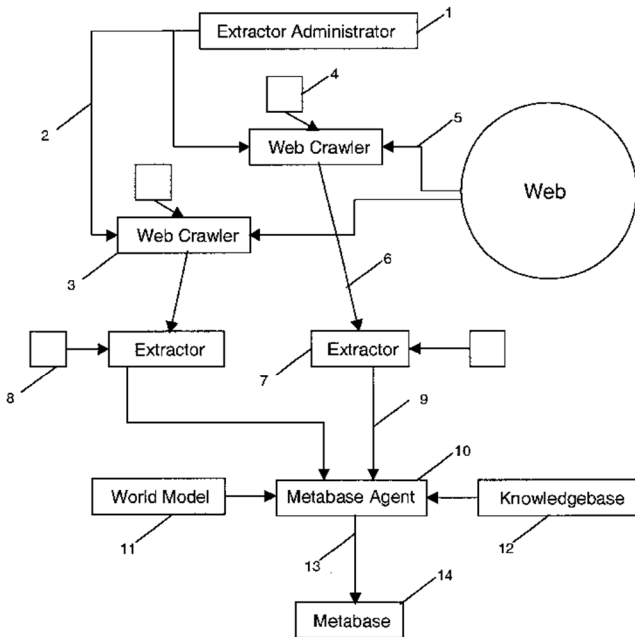
FIG. 5 shows the extraction process during which metadata for digital media is discovered, copied from a Web site, enhanced, and inserted into the present invention's database. Five modules contribute to this task:

- 1) WebCrawlers 3 that scan certain Web sites;
- 2) Extractors 7 that do the actual metadata extraction;

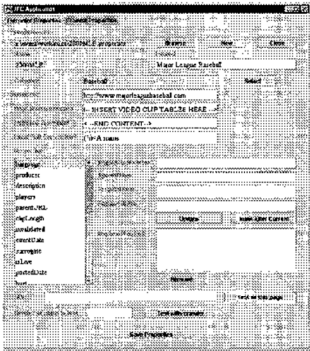
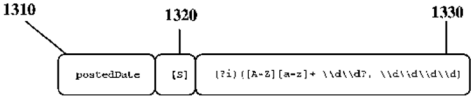
'796 Patent	<u>Sheth</u>
	3) Knowledgebase 12 that improves the metadata where possible; 4) Metabase Agent 10 that inserts the data into the present invention's metabase; and 5) Extractor Administrator 1 that schedules the running of WebCrawlers. The Extractor Administration 1 module maintains a list of all available crawlers/extractors along with a description of when the respective extraction process should be started and initiates extraction according to the schedule. The process starts by identifying a source containing digital media data sets. Several factors determine whether a source is actually worth extracting. The foregoing discussion refers to a Web site as an example source of digital media for retrieval; other potential sources such as a local drive or another remote resource such as FTP, GOPHER, etc. type material will be understood by those of skill in the art to be encompassed within the scope of the present invention. These include the volume of media files and the quality of the available metadata about those media files. Once a Web site is deemed worthy of undergoing this process, the assets that will be acquired from the site are assigned to a particular domain in the WorldModel. Next, crawling and extraction rules are written that specify where and how to retrieve the metadata from the Web site. For each attribute of a particular asset type (domain), an extractor writer provides a rule that specifies where to find a value of this attribute. Crawling and extraction rules are input to WebCrawler and Extractor programs that traverse Web sites and retrieve digital media metadata from selected pages. An extractor program takes HTML pages and extraction rules as input and generates XML assets such as that shown in FIG. 6. These generated assets contain values for each attribute name belonging to the domain of that Web site. Once created, the assets are sent to a Metabase Agent that is in charge of enhancing and inserting them into a database of records. In order to enhance the assets, the Metabase Agent uses information stored in the WorldModel as well as a Knowledgebase. The Knowledgebase is a collection of tables containing domain-specific information and relationships. After insertion into the metabase, the assets are then ready to be searched.

'796 Patent	<u>Sheth</u>
	... The purpose of a WebCrawler is to read in the content of “extractable” Web pages from a site and pass this content on to the Extractors. A WebCrawler 3 is a piece of software, invoked on a remote or local host, which begins reading pages from a particular site and determines which of these pages are extractable. When a WebCrawler is invoked 2, it is given a list of “crawling rules” 4 specific to a single Web site. These rules dictate where (on which page) the crawler should begin its search, which directories the crawler must remain within, and define the characteristics of an extractable page. Without such rules, a WebCrawler would likely find a link off of the site it was assigned to crawl and begin aimlessly reading the entire Web. ... Extractors 7 are programs that are designed to find information about digital media from a Web page. Their ability to retrieve values for domain-specific attributes is critical to capturing the semantics of the information on the source site. Extractors exploit Web pages that follow a template or pages that are generated by server-side scripts/programs. By assuming that Web pages from a particular site follow a pattern or have some recognizable structure, rules can be written that reliably retrieve small sections of text from those pages. ... When a WebCrawler finds an extractable page it will send this page (as well as the name of the site on which it is crawling) to one extractor in a large pool of extractors running on another machine. An extractor is designed to work on only one of these pages a time. Once an extractor receives a page, it looks up a set of extraction rules 8 associated with the site from which the page came. These rules list the metadata attributes for the type of media that this site contains as well as rules that describe where to find values for these attributes within the page. The set of attributes associated with, for example, a news video (reporter, location, event date, etc.) is different from the set of attributes associated with a sports highlight (teams, players, score etc.) The extractor scans the Web page content for pieces of text that match the

'796 Patent	<u>Sheth</u>
	pattern specified by the extraction rules. After the extractor has attempted to find every attribute in its list of extraction rules, it creates an XML document containing attribute-value pairs 9. This document is sent to the Metabase Agent 10. ... Referring back to FIG. 5, the role of the Metabase Agent 10 is to take XML assets, enhance them using the WorldModel 11 and the Knowledgebase 12, and insert the assets into the metabase of the present invention. The XML document that the Metabase Agent 10 receives contains the asset type and a list of attribute-value pairs. As an example, a particular asset extracted from CNN.com could have the type "News Asset", location "Atlanta", clip length "1:32", as well as several other attribute-values. ... Once the functions have been applied, the Metabase Agent will use the MediaAnywhere Knowledgebase 12 to further enhance the extracted information. In the case of the location "Atlanta", the Metabase Agent would consult a database table of world cities to expand the location to "Atlanta: Georgia: United States: North America". Thus, our sample asset can be retrieved by a location search for "Atlanta", "Georgia", "United States", and "North America". Finally, after all the mappings and enhancements have taken place, the Metabase Agent inserts the XML asset into the metabase. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other

'796 Patent	<u>Sheth</u>
[1A] identifying at least one regularly identifiable expression in the input sequence of data symbols, wherein the at least one regularly identifiable expression represents a pattern that is matchable in accordance with a programming language that supports such a regularly identifiable expression;	prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart. <u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation. <i>See, e.g.,</i> FIG. 5, 10:7-18:  Fig. 5 FIG. 5 shows the extraction process during which metadata for digital media is discovered, copied from a Web site, enhanced, and inserted into the present invention's database. Five modules contribute to this task:

'796 Patent	<u>Sheth</u>								
	relationships. After insertion into the metabase, the assets are then ready to be searched. ... Extractors 7 are programs that are designed to find information about digital media from a Web page. Their ability to retrieve values for domain-specific attributes is critical to capturing the semantics of the information on the source site. Extractors exploit Web pages that follow a template or pages that are generated by server-side scripts/programs. By assuming that Web pages from a particular site follow a pattern or have some recognizable structure, rules can be written that reliably retrieve small sections of text from those pages. Normally, these small pieces of text have some relationship to a media file whose link is on the same page. For example, a piece of significant text may contain "06/01/2000", "Barbados, Caribbean", or "CAB". In order for this information to be meaningful, they need to be mapped to a domain-specific attribute in the WorldModel. For example: <table border="1" data-bbox="821 808 1388 954"> <thead> <tr> <th>Extracted text</th><th>Domain-Specific Attribute</th></tr> </thead> <tbody> <tr> <td>06/01/2000</td><td>News (event date)</td></tr> <tr> <td>Barbados, Caribbean</td><td>Travel (location)</td></tr> <tr> <td>CAB</td><td>Business (stock symbol)</td></tr> </tbody> </table> This mapping is accomplished when a human writes a set of extraction rules for a site. When a WebCrawler finds an extractable page it will send this page (as well as the name of the site on which it is crawling) to one extractor in a large pool of extractors running on another machine. An extractor is designed to work on only one of these pages a time. Once an extractor receives a page, it looks up a set of extraction rules 8 associated with the site from which the page came. These rules list the metadata attributes for the type of media that this site contains as well as rules that describe where to find values for these attributes within the page. The set of attributes associated with, for example, a news video (reporter, location, event date, etc.) is different from the set of attributes associated with a sports highlight (teams, players,	Extracted text	Domain-Specific Attribute	06/01/2000	News (event date)	Barbados, Caribbean	Travel (location)	CAB	Business (stock symbol)
Extracted text	Domain-Specific Attribute								
06/01/2000	News (event date)								
Barbados, Caribbean	Travel (location)								
CAB	Business (stock symbol)								

'796 Patent	Sheth
	score etc.) The extractor scans the Web page content for pieces of text that match the pattern specified by the extraction rules. After the extractor has attempted to find every attribute in its list of extraction rules, it creates an XML document containing attribute-value pairs 9. This document is sent to the Metabase Agent 10. An example of such an XML document is shown in FIG. 6. FIGS. 7-8, 11:57-67:  Fig. 7 <pre># Name : CNNUS # producer [e]CNN description [l](?i)]*alt="Video"[^>]*>{[<]*}< descriptionReplace1 s/\\([\\ \\]*\\ \\ parentURL [u](.*) invalidated [e]0 surrogate [l](?i){[<]*}</h> productionSource [e]tv location [S](?i)/date.*?>(.*)\\ (.*)\\ locationReplace1 u/.*<\\>.*?<\\ >(.*)/S/ locationReplace2 s/(.*)?--.*?/S/ locationReplace3 s/.?.*?<\\ emo(.*)/S/ mediaType [e]video source [e]http://www.cnn.com</pre> Fig. 8  Fig. 9 Extraction rules for a site can usually be created in five minutes to an hour (depending on the complexity and irregularity of the site) using an Extractor Toolkit such as that shown in FIG. 7. This toolkit allows a relatively non-technical person to write extraction rules, such as those shown in FIG. 8, that pick out relevant information from a Web page and map this information to attributes in the WorldModel. The ability to retrieve text that matches a particular pattern is provided by the PERL

'796 Patent	<u>Sheth</u>
	regular expression language. To this language, the present invention adds several important features. Ability to modify extracted text (append, prepend, replace) Ability to split a page into one or more chunks that represent distinct logical assets Mechanism to allow page chunks to share common text Mechanism to allow metadata for one asset to be extracted from multiple pages Ability to decide (at runtime) the category of the assets on a given page The basic procedure for writing an extraction rule begins with choosing an attribute to extract. A list of possible attributes is displayed in the toolkit for a given category of asset (i.e. Baseball, News, Technology). The extractor author then tries to visually determine a pattern that describes either the location or the style of the text to be retrieved. For example, the desired text may be the first bold text on the page, the text after a particular picture, or the last URL link on the page. The author then examines the HTML sources of several similar pages to verify that such rules will consistently retrieve the correct text. Finally, the author types the rule for an attribute and can test it against several different pages. An example of a typical extraction rule is given in FIG. 9. Each extraction rule will contain potentially three components. The first component 1310 designates the name of the attribute. The second component 1320 indicates whether multiple assets are generated from a single data set subject to extraction and whether the attribute will be found in the common (shared) text, not the text belonging to the individual assets. The third component 1330 designates the pattern for which the extractor should search to locate the value for the attribute designated in the first component 1310. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element

'796 Patent	<u>Sheth</u>
	would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
[1B] identifying at least a portion of information associated with the at least one regularly identifiable expression; and	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation. <i>See, e.g.</i>, FIG. 5, 10:7-18: Fig. 5

'796 Patent	<u>Sheth</u>
	FIG. 5 shows the extraction process during which metadata for digital media is discovered, copied from a Web site, enhanced, and inserted into the present invention's database. Five modules contribute to this task: 1) WebCrawlers 3 that scan certain Web sites; 2) Extractors 7 that do the actual metadata extraction; 3) Knowledgebase 12 that improves the metadata where possible; 4) Metabase Agent 10 that inserts the data into the present invention's metabase; and 5) Extractor Administrator 1 that schedules the running of WebCrawlers. FIGS. 5-6, 10:23-54, 11:11-56: <pre> <?xml version="1.0"?> <DOCTYPE Asset [<ELEMENT Asset (category, attribute)*> <ELEMENT category (#PCDATA)> <ELEMENT attribute (#PCDATA)> <ATTLIST attribute name CDATA #REQUIRED>]> <Asset> <category>Movie</category> <attribute name="description">Remember rubik&apos;s cubes and parachute pants? Adam Sandler does. And he&apos;s more than happy...</attribute> <attribute name="producer">Hollywood</attribute> <attribute name="parentUrl">http://www.hollywood.com/video/movies/weddingsinger/index.html</attribute> <attribute name="urlpage">http://www.hollywood.com/video/movies/weddingsinger/index.html</attribute> <attribute name="credits">Directed by Frank Coraci. Written by Tim Herlihy. Produced by Jack Giarraputo, Michelle Hoddsworth...</attribute> <attribute name="surrogate">http://www.hollywood.com/movleguide/movies/weddingsinger/horizontal.jpg</attribute> <attribute name="isLive">0</attribute> <attribute name="cast">Adam Sandler, Drew Barrymore, Christine Taylor, Matthew Glave, Angela Featherstone, Allen Covert, Steve Buscemi, Ellen Alberini...</attribute> <attribute name="title">The Wedding Singer</attribute> <attribute name="productionSource">tv</attribute> <attribute name="genre">Comedy</attribute> <attribute name="mediaType">video</attribute> <attribute name="rating">PG-13, for sex-related material and language</attribute> <attribute name="source">http://www.hollywood.com</attribute> <attribute name="url">http://www.hollywood.com/trailers/weddingsinger/weddingsinger.mov</attribute> <attribute name="url">http://www.hollywood.com/trailers/weddingsinger/weddingsinger.vob</attribute> <attribute name="url">http://www.hollywood.com/trailers/weddingsinger/low_weddingsinger.asx</attribute> <attribute name="url">http://www.hollywood.com/trailers/weddingsinger/high_weddingsinger.asx</attribute> </Asset> </pre> Fig. 6 An extractor program takes HTML pages and extraction rules as input and generates XML assets such as that shown in FIG. 6. These generated assets contain values for each attribute name belonging to the domain of that Web site. Once created, the assets are sent to a Metabase Agent that is in charge of enhancing and inserting them into a database of records. In order to enhance the assets, the Metabase Agent uses

'796 Patent	<u>Sheth</u>								
	information stored in the WorldModel as well as a Knowledgebase. The Knowledgebase is a collection of tables containing domain-specific information and relationships. After insertion into the metabase, the assets are then ready to be searched. ... Extractors 7 are programs that are designed to find information about digital media from a Web page. Their ability to retrieve values for domain-specific attributes is critical to capturing the semantics of the information on the source site. Extractors exploit Web pages that follow a template or pages that are generated by server-side scripts/programs. By assuming that Web pages from a particular site follow a pattern or have some recognizable structure, rules can be written that reliably retrieve small sections of text from those pages. Normally, these small pieces of text have some relationship to a media file whose link is on the same page. For example, a piece of significant text may contain "06/01/2000", "Barbados, Caribbean", or "CAB". In order for this information to be meaningful, they need to be mapped to a domain-specific attribute in the WorldModel. For example: <table border="1" data-bbox="821 878 1388 1029"> <thead> <tr> <th>Extracted text</th><th>Domain-Specific Attribute</th></tr> </thead> <tbody> <tr> <td>06/01/2000</td><td>News (event date)</td></tr> <tr> <td>Barbados, Caribbean</td><td>Travel (location)</td></tr> <tr> <td>CAB</td><td>Business (stock symbol)</td></tr> </tbody> </table> This mapping is accomplished when a human writes a set of extraction rules for a site. When a WebCrawler finds an extractable page it will send this page (as well as the name of the site on which it is crawling) to one extractor in a large pool of extractors running on another machine. An extractor is designed to work on only one of these pages a time. Once an extractor receives a page, it looks up a set of extraction rules 8 associated with the site from which the page came. These rules list the metadata attributes for the type of media that this site contains as well as rules that describe where to find values for these attributes within the page. The set of attributes	Extracted text	Domain-Specific Attribute	06/01/2000	News (event date)	Barbados, Caribbean	Travel (location)	CAB	Business (stock symbol)
Extracted text	Domain-Specific Attribute								
06/01/2000	News (event date)								
Barbados, Caribbean	Travel (location)								
CAB	Business (stock symbol)								

'796 Patent	<u>Sheth</u>
	ability to retrieve text that matches a particular pattern is provided by the PERL regular expression language. To this language, the present invention adds several important features. Ability to modify extracted text (append, prepend, replace) Ability to split a page into one or more chunks that represent distinct logical assets Mechanism to allow page chunks to share common text Mechanism to allow metadata for one asset to be extracted from multiple pages Ability to decide (at runtime) the category of the assets on a given page The basic procedure for writing an extraction rule begins with choosing an attribute to extract. A list of possible attributes is displayed in the toolkit for a given category of asset (i.e. Baseball, News, Technology). The extractor author then tries to visually determine a pattern that describes either the location or the style of the text to be retrieved. For example, the desired text may be the first bold text on the page, the text after a particular picture, or the last URL link on the page. The author then examines the HTML sources of several similar pages to verify that such rules will consistently retrieve the correct text. Finally, the author types the rule for an attribute and can test it against several different pages. An example of a typical extraction rule is given in FIG. 9. Each extraction rule will contain potentially three components. The first component 1310 designates the name of the attribute. The second component 1320 indicates whether multiple assets are generated from a single data set subject to extraction and whether the attribute will be found in the common (shared) text, not the text belonging to the individual assets. The third component 1330 designates the pattern for which the extractor should search to locate the value for the attribute designated in the first component 1310. FIG. 9, 12:10-42: The basic procedure for writing an extraction rule begins with choosing an attribute to extract. A list of possible attributes is displayed in the toolkit for a given category of asset (i.e. Baseball, News, Technology). The extractor author then tries to visually

'796 Patent	<u>Sheth</u>
	determine a pattern that describes either the location or the style of the text to be retrieved. For example, the desired text may be the first bold text on the page, the text after a particular picture, or the last URL link on the page. The author then examines the HTML sources of several similar pages to verify that such rules will consistently retrieve the correct text. Finally, the author types the rule for an attribute and can test it against several different pages. An example of a typical extraction rule is given in FIG. 9. Each extraction rule will contain potentially three components. The first component 1310 designates the name of the attribute. The second component 1320 indicates whether multiple assets are generated from a single data set subject to extraction and whether the attribute will be found in the common (shared) text, not the text belonging to the individual assets. The third component 1330 designates the pattern for which the extractor should search to locate the value for the attribute designated in the first component 1310. Referring back to FIG. 5, the role of the Metabase Agent 10 is to take XML assets, enhance them using the WorldModel 11 and the Knowledgebase 12, and insert the assets into the metabase of the present invention. The XML document that the Metabase Agent 10 receives contains the asset type and a list of attribute-value pairs. As an example, a particular asset extracted from CNN.com could have the type "News Asset", location "Atlanta", clip length "1:32", as well as several other attribute-values. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.

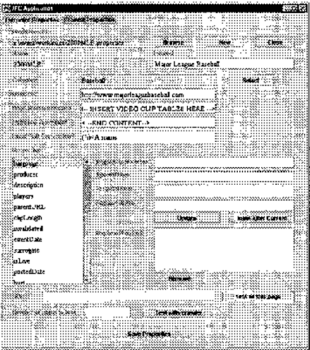
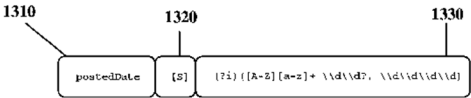
'796 Patent	<u>Sheth</u>
[1C] extracting the portion of information.	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation. <i>See, e.g.</i>, FIG. 5, 10:7-18: Fig. 5 FIG. 5 shows the extraction process during which metadata for digital media is discovered, copied from a Web site, enhanced, and inserted into the present invention's database. Five modules contribute to this task: 1) WebCrawlers 3 that scan certain Web sites; 2) Extractors 7 that do the actual metadata extraction;

'796 Patent	<u>Sheth</u>
	3) Knowledgebase 12 that improves the metadata where possible; 4) Metabase Agent 10 that inserts the data into the present invention's metabase; and 5) Extractor Administrator 1 that schedules the running of WebCrawlers. FIGS. 5-6, 10:23-54, 11:11-56: <pre> <?xml version="1.0"?> <!DOCTYPE Asset [<ELEMENT Asset (category, attribute)*> <ELEMENT category (#PCDATA)> <ELEMENT attribute (#PCDATA)> <ATTLIST attribute name CDATA #REQUIRED>]> <Asset> <category>Movie</category> <attribute name="description">Remember rubik&apos;s cubes and parachute pants? Adam Sandler docs. And he&apos;s more than happy...</attribute> <attribute name="producer">Hollywood</attribute> <attribute name="parentUri">http://www.hollywood.com/video/guide/movies/weddingsinger/index.html</attribute> <attribute name="urlpage">http://www.hollywood.com/video/guide/movies/weddingsinger/trailer/index.html</attribute> <attribute name="credit">Directed by Frank Coraci. Written by Tim Herlihy. Produced by Jack Giampato, Michelle Holdsworth...</attribute> <attribute name="thumbnail">http://www.hollywood.com/movleguide/movies/weddingsinger/horizontal.jpg</attribute> <attribute name="isLive">0</attribute> <attribute name="cast">Adam Sandler, Drew Barrymore, Christine Taylor, Matthew Glave, Angela Featherstone, Allen Covert, Steve Buscemi, Ellen Albertini...</attribute> <attribute name="title">The Wedding Singer</attribute> <attribute name="productionSource">tv</attribute> <attribute name="genre">Comedy</attribute> <attribute name="mediaType">video</attribute> <attribute name="rating">PG-13, for sex-related material and language</attribute> <attribute name="source">http://www.hollywood.com</attribute> <attribute name="url">http://www.hollywood.com/trailers/weddingsinger/weddingsinger.mov</attribute> <attribute name="url">http://www.hollywood.com/trailers/weddingsinger/weddingsinger.vob</attribute> <attribute name="url">http://www.hollywood.com/trailers/weddingsinger/low_weddingsinger.asx</attribute> <attribute name="url">http://www.hollywood.com/trailers/weddingsinger/high_weddingsinger.asx</attribute> </Asset> </pre> ...

Fig. 6

An extractor program takes HTML pages and extraction rules as input and generates XML assets such as that shown in FIG. 6. These generated assets contain values for each attribute name belonging to the domain of that Web site. Once created, the assets are sent to a Metabase Agent that is in charge of enhancing and inserting them into a database of records. In order to enhance the assets, the Metabase Agent uses information stored in the WorldModel as well as a Knowledgebase. The Knowledgebase is a collection of tables containing domain-specific information and relationships. After insertion into the metabase, the assets are then ready to be searched.

'796 Patent	<u>Sheth</u>								
	Extractors 7 are programs that are designed to find information about digital media from a Web page. Their ability to retrieve values for domain-specific attributes is critical to capturing the semantics of the information on the source site. Extractors exploit Web pages that follow a template or pages that are generated by server-side scripts/programs. By assuming that Web pages from a particular site follow a pattern or have some recognizable structure, rules can be written that reliably retrieve small sections of text from those pages. Normally, these small pieces of text have some relationship to a media file whose link is on the same page. For example, a piece of significant text may contain "06/01/2000", "Barbados, Caribbean", or "CAB". In order for this information to be meaningful, they need to be mapped to a domain-specific attribute in the WorldModel. For example: <table data-bbox="821 662 1388 816"> <tr> <th>Extracted text</th><th>Domain-Specific Attribute</th></tr> <tr> <td>06/01/2000</td><td>News (event date)</td></tr> <tr> <td>Barbados, Caribbean</td><td>Travel (location)</td></tr> <tr> <td>CAB</td><td>Business (stock symbol)</td></tr> </table> This mapping is accomplished when a human writes a set of extraction rules for a site. When a WebCrawler finds an extractable page it will send this page (as well as the name of the site on which it is crawling) to one extractor in a large pool of extractors running on another machine. An extractor is designed to work on only one of these pages a time. Once an extractor receives a page, it looks up a set of extraction rules 8 associated with the site from which the page came. These rules list the metadata attributes for the type of media that this site contains as well as rules that describe where to find values for these attributes within the page. The set of attributes associated with, for example, a news video (reporter, location, event date, etc.) is different from the set of attributes associated with a sports highlight (teams, players, score etc.) The extractor scans the Web page content for pieces of text that match the pattern specified by the extraction rules. After the extractor has attempted to find every attribute in its list of extraction rules, it creates an XML document containing	Extracted text	Domain-Specific Attribute	06/01/2000	News (event date)	Barbados, Caribbean	Travel (location)	CAB	Business (stock symbol)
Extracted text	Domain-Specific Attribute								
06/01/2000	News (event date)								
Barbados, Caribbean	Travel (location)								
CAB	Business (stock symbol)								

'796 Patent	Sheth
	attribute-value pairs 9. This document is sent to the Metabase Agent 10. An example of such an XML document is shown in FIG. 6. FIGS. 7-8, 11:57-67:  <pre># Name : CNNUS # producer [c]CNN description [l](<i>{[^<]*}< s/\[\{[^\\]**\}\\]/ parentURL [u](.*) invalidated [c]0 surrogate [l](<i><img src='(/video/.*)' surrogatePrepend [c]0 http://www.cnn.com postedDate [S](<i>{[A-Z][a-z]* \d\d\d\d?, \d\d\d\d\d\d\d) url [m](<i>javascript:vod\[\{(/video/{[^}]*', urlReplace1 s/\[\{[^\\]**\}\\]/html/pnm:/\[\{real\\\.cnn\\\.com\$irm/ urlReplace2 s/\[\{[^\\]**\}\\]/html/pnm:/\[\{real\\\.cnn\\\.com\$irm/ urlReplace3 s/\[\{[^\\]**\}\\]/html/pnm:/\[\{vod\\\.cnn\\\.com\$last/ urlReplace4 s/\[\{[^\\]**\}\\]/html/pnm:/\[\{vod\\\.cnn\\\.com\$last/ urlReplace5 s/\[\{[^\\]**\}\\]/html/http:/\[\{vod\\\.cnn.com\$last/ title [S](<i>ch.>{[^<]*}</h> productionSource [c]tv location [S](<i>{<date>.*?<sp>(.*)\\{(.*)\\} locationReplace1 u.*?<id>.*?<\\{>(.*)/\$1/ locationReplace2 s/(.*)--.*?/\$1/ locationReplace3 s/.?<acc>.*?<\\emo(.*)/\$1/ mediaType [c]video source [c]http://www.cnn.com</pre> Fig. 7 Fig. 8  Fig. 9 Extraction rules for a site can usually be created in five minutes to an hour (depending on the complexity and irregularity of the site) using an Extractor Toolkit such as that shown in FIG. 7. This toolkit allows a relatively non-technical person to write extraction rules, such as those shown in FIG. 8, that pick out relevant information from a Web page and map this information to attributes in the WorldModel. The ability to retrieve text that matches a particular pattern is provided by the PERL regular expression language. To this language, the present invention adds several important features. Ability to modify extracted text (append, prepend, replace)

'796 Patent	<u>Sheth</u>
	Ability to split a page into one or more chunks that represent distinct logical assets Mechanism to allow page chunks to share common text Mechanism to allow metadata for one asset to extracted from multiple pages Ability to decide (at runtime) the category of the assets on a given page The basic procedure for writing an extraction rule begins with choosing an attribute to extract. A list of possible attributes is displayed in the toolkit for a given category of asset (i.e. Baseball, News, Technology). The extractor author then tries to visually determine a pattern that describes either the location or the style of the text to be retrieved. For example, the desired text may be the first bold text on the page, the text after a particular picture, or the last URL link on the page. The author then examines the HTML sources of several similar pages to verify that such rules will consistently retrieve the correct text. Finally, the author types the rule for an attribute and can test it against several different pages. An example of a typical extraction rule is given in FIG. 9. Each extraction rule will contain potentially three components. The first component 1310 designates the name of the attribute. The second component 1320 indicates whether multiple assets are generated from a single data set subject to extraction and whether the attribute will be found in the common (shared) text, not the text belonging to the individual assets. The third component 1330 designates the pattern for which the extractor should search to locate the value for the attribute designated in the first component 1310. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other

'796 Patent	<u>Sheth</u>
	prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
[2] The method of claim 1, wherein the regularly identifiable expression identifying step comprises comparing the input sequence of data symbols to one or more previously-stored regularly identifiable expressions to determine if there is a match between a portion of the input sequence and at least one of the previously-stored regularly identifiable expressions.	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation. <i>See, e.g.,</i> FIGS. 5-6, 10:23-11:10: The process starts by identifying a source containing digital media data sets. Several factors determine whether a source is actually worth extracting. The foregoing discussion refers to a Web site as an example source of digital media for retrieval; other potential sources such as a local drive or another remote resource such as FTP, GOPHER, etc. type material will be understood by those of skill in the art to be encompassed within the scope of the present invention. These include the volume of media files and the quality of the available metadata about those media files. Once a Web site is deemed worthy of undergoing this process, the assets that will be acquired from the site are assigned to a particular domain in the WorldModel. Next, crawling and extraction rules are written that specify where and how to retrieve the metadata from the Web site. For each attribute of a particular asset type (domain), an extractor writer provides a rule that specifies where to find a value of this attribute. Crawling and extraction rules are input to WebCrawler and Extractor programs that traverse Web sites and retrieve digital media metadata from selected pages. An extractor program takes HTML pages and extraction rules as input and generates XML assets such as that shown in FIG. 6. These generated assets contain values for each attribute name belonging to the domain of that Web site. Once created, the assets are sent to a Metabase Agent that is in charge of enhancing and inserting them into a database of records. In order to enhance the assets, the Metabase Agent uses information stored in the WorldModel as well as a Knowledgebase. The Knowledgebase is a collection of tables containing domain-specific information and relationships. After insertion into the metabase, the assets are then ready to be searched.

'796 Patent	<u>Sheth</u>
	The purpose of a WebCrawler is to read in the content of “extractable” Web pages from a site and pass this content on to the Extractors. A WebCrawler 3 is a piece of software, invoked on a remote or local host, which begins reading pages from a particular site and determines which of these pages are extractable. When a WebCrawler is invoked 2, it is given a list of “crawling rules” 4 specific to a single Web site. These rules dictate where (on which page) the crawler should begin its search, which directories the crawler must remain within, and define the characteristics of an extractable page. Without such rules, a WebCrawler would likely find a link off of the site it was assigned to crawl and begin aimlessly reading the entire Web. An “extractable page” 5 is a Web page that both follows a template (however loosely) and likely contains a link to a media asset. A crawler can often recognize an extractable page by simply examining the URL. If the URL of the page being examined follows a certain pattern, there is a good chance that the page contains a link to an audio or video asset. When the WebCrawler determines that it has found an extractable page, it sends the contents of that page 6 on to another remote piece of software (an Extractor) that retrieves the valuable metadata from the page. Any number of WebCrawlers can run on any host that is set up to do so. FIG. 9, 12:10-33: The basic procedure for writing an extraction rule begins with choosing an attribute to extract. A list of possible attributes is displayed in the toolkit for a given category of asset (i.e. Baseball, News, Technology). The extractor author then tries to visually determine a pattern that describes either the location or the style of the text to be retrieved. For example, the desired text may be the first bold text on the page, the text after a particular picture, or the last URL link on the page. The author then examines the HTML sources of several similar pages to verify that such rules will consistently retrieve the correct text. Finally, the author types the rule for an attribute and can test it against several different pages. An example of a typical extraction rule is given in FIG. 9. Each extraction rule will contain potentially three components. The first component 1310 designates the name

'796 Patent	<u>Sheth</u>
	of the attribute. The second component 1320 indicates whether multiple assets are generated from a single data set subject to extraction and whether the attribute will be found in the common (shared) text, not the text belonging to the individual assets. The third component 1330 designates the pattern for which the extractor should search to locate the value for the attribute designated in the first component 1310. FIG. 10, 12:34-13:27: Referring back to FIG. 5, the role of the Metabase Agent 10 is to take XML assets, enhance them using the WorldModel 11 and the Knowledgebase 12, and insert the assets into the metabase of the present invention. The XML document that the Metabase Agent 10 receives contains the asset type and a list of attribute-value pairs. As an example, a particular asset extracted from CNN.com could have the type "News Asset", location "Atlanta", clip length "1:32", as well as several other attribute-values. In order to improve the quality of the data for this asset and enable higher recall, the Metabase Agent must consult the WorldModel. The WorldModel 11, which contains domain-dependent information for each type of asset in the metabase, can be implemented as a collection of XML documents or as a collection of tables in a relational database. One piece of information that the WorldModel contains is the name of a mapping function used to convert the format of an extracted value into a standard format. One simple mapping function would be the one that converts a clip length in minutes:seconds to seconds (the format accepted by the metabase). Other mapping functions could be used to map movie ratings from several different sites to a common rating scheme. The Metabase Agent will find the names of the attributes and the mapping functions that should be applied from the WorldModel. Once the functions have been applied, the Metabase Agent will use the MediaAnywhere Knowledgebase 12 to further enhance the extracted information. In the case of the location "Atlanta", the Metabase Agent would consult a database table of world cities to expand the location to "Atlanta: Georgia: United States: North America". Thus, our sample asset can be retrieved by a location search for "Atlanta", "Georgia", "United States", and "North America". Finally, after all the mappings and

'796 Patent	<u>Sheth</u>
	enhancements have taken place, the Metabase Agent inserts the XML asset into the metabase. An example of asset enhancement is illustrated in FIG. 10. In this diagram, an extractor applies its extraction rules to an incoming HTML file. It then creates an asset object 31 containing a set of attribute-value pairs. In this example, the extractor is unable to find values for the “teams” and “players” attributes of a baseball asset. This is due to the fact that the task of reliably finding proper names in continuous text is nearly impossible using only a regular expression-based language. The extractor passes this incomplete asset on to the Metabase Agent. The Metabase Agent realizes that the asset is incomplete and consults the WorldModel 32 to look up an appropriate method of enhancement. In this case, the WorldModel informs the Metabase Agent that the “teams” and “players” attributes of baseball assets can be completed by looking in the title or description. The Metabase Agent then invokes an asset-enhancement routine that examines the title and description word-by-word and looks for teams or players in baseball knowledge tables 33. Not only does the Knowledgebase maintain lists of things such as teams and players, but it also stores any relationships between them (i.e. a “plays for” relationship). Even if the team name were unavailable from the description or title, it would still be found using the relationships stored in the Knowledgebase. After the enhancement phase is finished, the enhanced asset 34 is inserted into the Metabase. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.

EXHIBIT A-Sheth194

'796 Patent	<u>Sheth</u>

EXHIBIT A-Sheth194

'796 Patent	<u>Sheth</u>

EXHIBIT A-Sheth194

'796 Patent	<u>Sheth</u>

EXHIBIT A-Sheth194

'796 Patent	<u>Sheth</u>
[7] The method of claim 1, wherein the extracted portion of information is used to take a specified action.	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation. <i>See, e.g.</i> ,

'796 Patent	Sheth
	10:43-54: An extractor program takes HTML pages and extraction rules as input and generates XML assets such as that shown in FIG. 6. These generated assets contain values for each attribute name belonging to the domain of that Web site. Once created, the assets are sent to a Metabase Agent that is in charge of enhancing and inserting them into a database of records. In order to enhance the assets, the Metabase Agent uses information stored in the WorldModel as well as a Knowledgebase. The Knowledgebase is a collection of tables containing domain-specific information and relationships. After insertion into the metabase, the assets are then ready to be searched. FIG. 6, 11:52-56: <pre> <?xml version="1.0"?> <!DOCTYPE Asset [<ELEMENT Asset (category, attribute)*> <ELEMENT category (#PCDATA)> <ELEMENT attribute (#PCDATA)> <ATTLIST attribute name CDATA #REQUIRED>]> <Asset> <category>Movie</category> <attribute name="description">Remember rubik&apos;s cubes and parachute pants? Adam Sandler does. And he&apos;s more than happy...</attribute> <attribute name="producer">Hollywood</attribute> <attribute name="parentUrl">http://www.hollywood.com/video/movies/weddingsinger/index.html</attribute> <attribute name="urlpage">http://www.hollywood.com/video/movies/weddingsinger/trailer/index.html</attribute> <attribute name="credits">Directed by Frank Coraci. Written by Tim Herlihy. Produced by Jack Giannapulo, Michelle Hodsaworth...</attribute> <attribute name="surrogate">http://www.hollywood.com/movleguide/movies/weddingsinger/horizontal.jpg</attribute> <attribute name="isLive">0</attribute> <attribute name="cast">Adam Sandler, Drew Barrymore, Christine Taylor, Matthew Glave, Angela Featherstone, Allen Covert, Steve Buscemi, Ellen Albertini...</attribute> <attribute name="title">The Wedding Singer</attribute> <attribute name="productionSource">tv</attribute> <attribute name="genre">Comedy</attribute> <attribute name="mediaType">video</attribute> <attribute name="rating">PG-13, for sex-related material and language</attribute> <attribute name="source">http://www.hollywood.com</attribute> <attribute name="url">http://www.hollywood.com/trailers/weddingsinger/weddingsinger.mov</attribute> <attribute name="url">http://www.hollywood.com/trailers/weddingsinger/weddingsinger.vob</attribute> <attribute name="url">http://www.hollywood.com/trailers/weddingsinger/low_weddingsinger.asx</attribute> <attribute name="url">http://www.hollywood.com/trailers/weddingsinger/high_weddingsinger.asx</attribute> </Asset> </pre> Fig. 6 After the extractor has attempted to find every attribute in its list of extraction rules, it creates an XML document containing attribute-value pairs 9. This document is sent to the Metabase Agent 10. An example of such an XML document is shown in FIG. 6.

'796 Patent

Sheth

FIG. 10, 12:34-13:41:

FOXSports.com

File Edit View Favorites Tools Help

Address http://www.foxsports.com/media/000308/0308arod.xml

A-Rod not going anywhere

The Seattle Mariners said Wednesday they will not trade All-Star shortstop Alex Rodriguez this season. Rodriguez says he's excited about the year ahead and GM Pat Gillick says the Mariners had no intentions of trading him this season.

Baseball Asset

Title	A-Rod not going anywhere
Description	The Seattle Mariners said...
Surrogate	http://www.foxsports/images/..
Video URL	http://www.foxsports/video/..
Parent URL	http://www.foxsports/media/..
•	
•	
Players	Alex Rodriguez
Teams	Seattle Mariners

Extractor

Extraction Rules

Baseball Asset

Title	A-Rod not going anywhere
Description	The Seattle Mariners said...
Surrogate	http://www.foxsports/images/..
Video URL	http://www.foxsports/video/..
Parent URL	http://www.foxsports/media/..
•	
•	
Players	???
Teams	???

World Model

Metabase Agent

Knowledgebase

31

32

33

34

Fig. 10

Referring back to FIG. 5, the role of the Metabase Agent 10 is to take XML assets, enhance them using the WorldModel 11 and the Knowledgebase 12, and insert the assets into the metabase of the present invention. The XML document that the Metabase Agent 10 receives contains the asset type and a list of attribute-value pairs. As an example, a particular asset extracted from CNN.com could have the type “News Asset”, location “Atlanta”, clip length “1:32”, as well as several other attribute-values.

In order to improve the quality of the data for this asset and enable higher recall, the Metabase Agent must consult the WorldModel. The WorldModel 11, which contains domain-dependent information for each type of asset in the metabase, can be

34

Taasera's Exhibit No. 2003, IPR2023-00443
Page 034

'796 Patent	<u>Sheth</u>
	implemented as a collection of XML documents or as a collection of tables in a relational database. One piece of information that the WorldModel contains is the name of a mapping function used to convert the format of an extracted value into a standard format. One simple mapping function would be the one that converts a clip length in minutes:seconds to seconds (the format accepted by the metabase). Other mapping functions could be used to map movie ratings from several different sites to a common rating scheme. The Metabase Agent will find the names of the attributes and the mapping functions that should be applied from the WorldModel. Once the functions have been applied, the Metabase Agent will use the MediaAnywhere Knowledgebase 12 to further enhance the extracted information. In the case of the location "Atlanta", the Metabase Agent would consult a database table of world cities to expand the location to "Atlanta: Georgia: United States: North America". Thus, our sample asset can be retrieved by a location search for "Atlanta", "Georgia", "United States", and "North America". Finally, after all the mappings and enhancements have taken place, the Metabase Agent inserts the XML asset into the metabase. An example of asset enhancement is illustrated in FIG. 10. In this diagram, an extractor applies its extraction rules to an incoming HTML file. It then creates an asset object 31 containing a set of attribute-value pairs. In this example, the extractor is unable to find values for the "teams" and "players" attributes of a baseball asset. This is due to the fact that the task of reliably finding proper names in continuous text is nearly impossible using only a regular expression-based language. The extractor passes this incomplete asset on to the Metabase Agent. The Metabase Agent realizes that the asset is incomplete and consults the WorldModel 32 to look up an appropriate method of enhancement. In this case, the WorldModel informs the Metabase Agent that the "teams" and "players" attributes of baseball assets can be completed by looking in the title or description. The Metabase Agent then invokes an asset-enhancement routine that examines the title and description word-by-word and looks for teams or players in baseball knowledge tables 33. Not only does the Knowledgebase maintain lists of things such as teams and players, but it also stores any relationships between them (i.e. a "plays for" relationship). Even if the team

'796 Patent	<u>Sheth</u>
	name were unavailable from the description or title, it would still be found using the relationships stored in the Knowledgebase. After the enhancement phase is finished, the enhanced asset 34 is inserted into the Metabase. Although not truly a part of the extraction process, quality assurance is an important part of ensuring that search requests lead to high-quality results. While most of the metadata acquisition happens automatically, human input may be required in some embodiments to resolve ambiguities (e.g., what does the location "Athens" expand to?). A LinkChecker module periodically goes through all stored URLs and tries to verify the validity of the link. For technical reasons (some audio/video formats are proprietary) it is not always possible to determine whether a particular media file is still accessible, but the LinkChecker can in many cases find "dead" links and flag them in the metabase so that the number of annoying "page not found" browser messages is reduced. 15:38-16:55: Companies like "Be Free" try to infer user preferences by gathering a great number of data about a user's browsing and viewing habits, including which sites were visited, how long, what banner ads were clicked, whether any ad lead to a purchase and many more. The present invention enables semantic profiling of the users of the services; user preferences can easily be tracked since the context of interest is provided with every step of the BSBQ. Furthermore, user profiles are managed as part of the WorldModel. Just as the assets can be cataloged according to the WorldModel, user profiles can also be cataloged according to the WorldModel. In order to show banner ads to site visitors that are likely to be responsive to such ads, it is vital to know as much about the user's interests as possible. The methods and apparatus of the present invention allow for semantic and immediate interest targeting. Those methods can instantaneously select and display commercial advertisements based on the semantic context that the Web user is currently browsing. Most advertisement targeting today is done using keywords. Statistics are typically gathered about the number of times a user visits a given page or the list of keywords for which a user often searches. However, such statistics contain no data

'796 Patent	<u>Sheth</u>
	on the context of the search terms. If the user enters “Michael Jordan” as a search keyword, there is no mechanism in the ad targeting system to recognize that Michael Jordan was basketball player for the Chicago Bulls. It only knows that the user has entered a two-word search term that appears in the description of some advertisements. Furthermore, advertisements are either managed by the system as part of the WorldModel, or by a third party, in which case the advertisements are mapped to the WorldModel. Just as the assets and user profiles can be cataloged according to the WorldModel, advertisements can also be cataloged according to the WorldModel. The present invention allows for two types of semantic advertising. The first type involves no external database of advertisements such as the one maintained by a partner company. The database of advertisements would be completely internal and would categorize ads into the same domains as the WorldModel. Similar to a media asset, each advertisement would have a set of attribute-value pairs that are unique to its domain. When the user queries the metabase for media assets, a separate advertisement query can be generated. For example, if a user looks for “Michael Jordan” in the domain of basketball, a sequence of queries can be performed to determine an advertisement that best matches the user's interest. The first and most precise query would involve searching for Michael Jordan basketball ads (perhaps footwear or autographs). This query mirrors the AND query that is being performed against the assets in the metabase. Similarly, OR queries can be performed if the precise AND query returns no results. Queries involving the Knowledgebase can also be useful. The Knowledgebase, in addition to storing asset-enhancing information, also stores relationships such as a player belonging to a team. In this case the Knowledgebase could determine that Michael Jordan played for the team “Chicago Bulls”. A query could also be run for Chicago Bull basketball ads (team jerseys, tickets) if none of the more precise ad queries returned results. Still, if no ad can be found, the domain hierarchy can be traversed backwards, performing a keyword query for “Michael Jordan” at each domain until a result is found. As an alternative, a generic basketball or sports ad could be used. In any case, the knowledge obtained

'796 Patent	<u>Sheth</u>
	from the user in the process of BSBQ preserves the semantics of ad request and produces results that are just as relevant as the media assets. The second form of targeted advertising would involve creating an XML string that represents queries that the user performs or individual media assets in which the user has an interest. This XML string could then be sent to an external advertisement provider along with a user's session and profile information. The advertisement provider would process this information and return a URL reference to a semantically targeted advertisement. Not only could such information be sent to advertisement providers, but similar information could also be sent to content providers. When users choose to play an audio or video file in a query result, a player for the media as well as an additional Web browser instance is created. The new browser instance would contain the Web page from which the asset was extracted. Taking the user back to the original source of the video or audio is a requirement for one-click media play. If the content provider is sent an XML string similar to the one sent to the ad provider, the content provider is able to tailor the advertisements in their own page (in the new browser instance). One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by
	reference into this chart.

EXHIBIT A-Sheth194

'796 Patent	<u>Sheth</u>

EXHIBIT A-Sheth194

'796 Patent	<u>Sheth</u>

EXHIBIT A-Sheth194

'796 Patent	<u>Sheth</u>

EXHIBIT A-Sheth194

'796 Patent	<u>Sheth</u>

EXHIBIT A-Sheth194

'796 Patent	<u>Sheth</u>

EXHIBIT A-Sheth194

'796 Patent	<u>Sheth</u>

'796 Patent	<u>Sheth</u>

EXHIBIT A-Sheth194

'796 Patent	<u>Sheth</u>

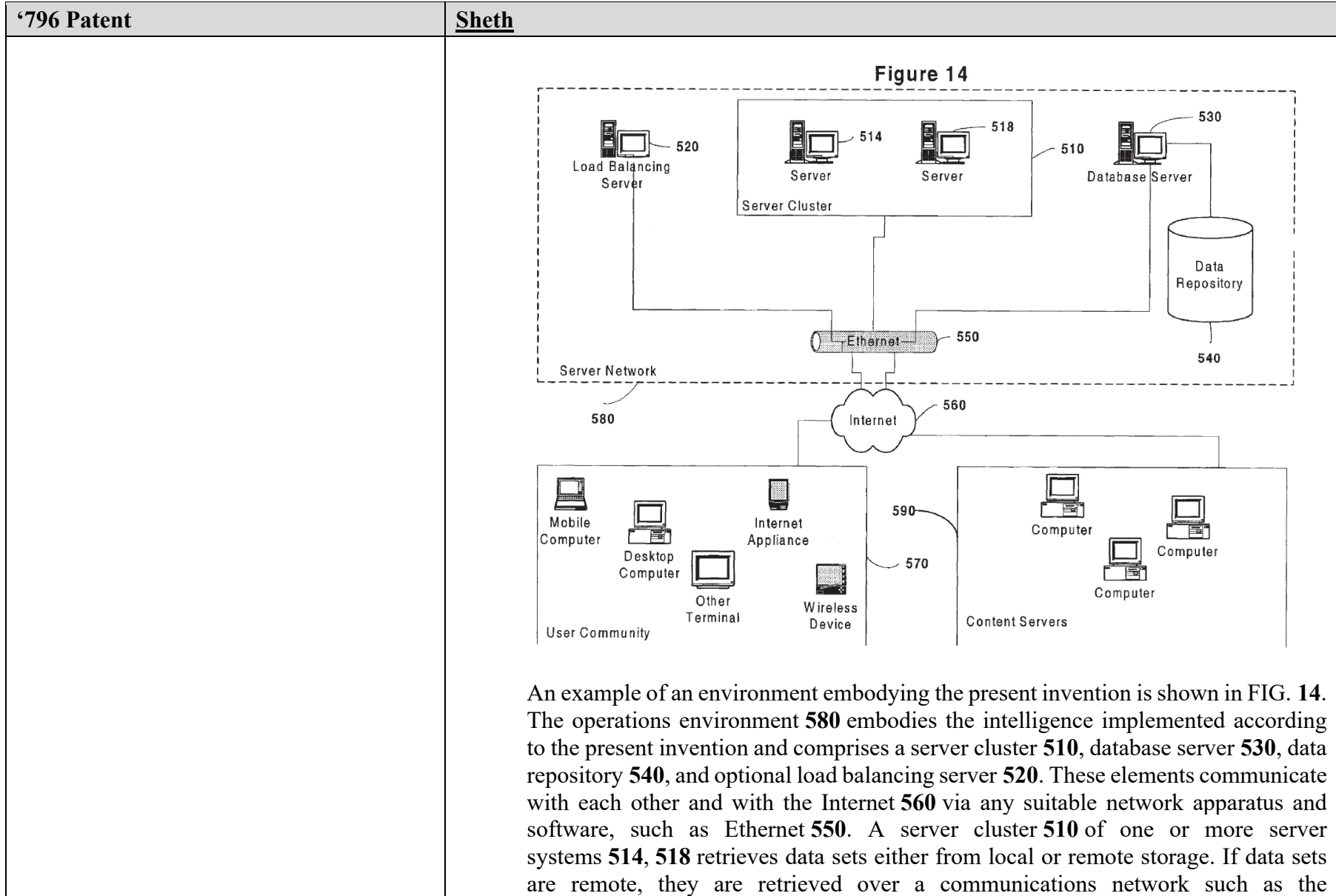
'796 Patent	<u>Sheth</u>
[10] The method of claim 1, wherein the input sequence of data symbols is representative of at least one of text data, transcribed spoken data, deoxyribonucleic acid sequence data, ribonucleic acid sequence data, amino-acid sequence data, audio sequence data, and video sequence data.	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation. <i>See, e.g.</i>, 10:43-11:56: An extractor program takes HTML pages and extraction rules as input and generates XML assets such as that shown in FIG. 6. These generated assets contain values for each attribute name belonging to the domain of that Web site. Once created, the assets are sent to a Metabase Agent that is in charge of enhancing and inserting them into a database of records. In order to enhance the assets, the Metabase Agent uses

'796 Patent	<u>Sheth</u>
	information stored in the WorldModel as well as a Knowledgebase. The Knowledgebase is a collection of tables containing domain-specific information and relationships. After insertion into the metabase, the assets are then ready to be searched. The purpose of a WebCrawler is to read in the content of “extractable” Web pages from a site and pass this content on to the Extractors. A WebCrawler 3 is a piece of software, invoked on a remote or local host, which begins reading pages from a particular site and determines which of these pages are extractable. When a WebCrawler is invoked 2, it is given a list of “crawling rules” 4 specific to a single Web site. These rules dictate where (on which page) the crawler should begin its search, which directories the crawler must remain within, and define the characteristics of an extractable page. Without such rules, a WebCrawler would likely find a link off of the site it was assigned to crawl and begin aimlessly reading the entire Web. An “extractable page” 5 is a Web page that both follows a template (however loosely) and likely contains a link to a media asset. A crawler can often recognize an extractable page by simply examining the URL. If the URL of the page being examined follows a certain pattern, there is a good chance that the page contains a link to an audio or video asset. When the WebCrawler determines that it has found an extractable page, it sends the contents of that page 6 on to another remote piece of software (an Extractor) that retrieves the valuable metadata from the page. Any number of WebCrawlers can run on any host that is set up to do so. Extractors 7 are programs that are designed to find information about digital media from a Web page. Their ability to retrieve values for domain-specific attributes is critical to capturing the semantics of the information on the source site. Extractors exploit Web pages that follow a template or pages that are generated by server-side scripts/programs. By assuming that Web pages from a particular site follow a pattern or have some recognizable structure, rules can be written that reliably retrieve small sections of text from those pages. Normally, these small pieces of text have some relationship to a media file whose link is on the same page. For example, a piece of significant text may contain “06/01/2000”, “Barbados, Caribbean”, or “CAB”. In

'796 Patent	<u>Sheth</u>								
	order for this information to be meaningful, they need to be mapped to a domain-specific attribute in the WorldModel. For example: <table border="1" data-bbox="835 360 1570 527"> <thead> <tr> <th>Extracted text</th><th>Domain-Specific Attribute</th></tr> </thead> <tbody> <tr> <td>06/01/2000</td><td>News (event date)</td></tr> <tr> <td>Barbados, Caribbean</td><td>Travel (location)</td></tr> <tr> <td>CAB</td><td>Business (stock symbol)</td></tr> </tbody> </table> This mapping is accomplished when a human writes a set of extraction rules for a site. When a WebCrawler finds an extractable page it will send this page (as well as the name of the site on which it is crawling) to one extractor in a large pool of extractors running on another machine. An extractor is designed to work on only one of these pages a time. Once an extractor receives a page, it looks up a set of extraction rules 8 associated with the site from which the page came. These rules list the metadata attributes for the type of media that this site contains as well as rules that describe where to find values for these attributes within the page. The set of attributes associated with, for example, a news video (reporter, location, event date, etc.) is different from the set of attributes associated with a sports highlight (teams, players, score etc.) The extractor scans the Web page content for pieces of text that match the pattern specified by the extraction rules. After the extractor has attempted to find every attribute in its list of extraction rules, it creates an XML document containing attribute-value pairs 9. This document is sent to the Metabase Agent 10. An example of such an XML document is shown in FIG. 6. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common	Extracted text	Domain-Specific Attribute	06/01/2000	News (event date)	Barbados, Caribbean	Travel (location)	CAB	Business (stock symbol)
Extracted text	Domain-Specific Attribute								
06/01/2000	News (event date)								
Barbados, Caribbean	Travel (location)								
CAB	Business (stock symbol)								

'796 Patent	<u>Sheth</u>
	sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.

'796 Patent	<u>Sheth</u>
[12P] Apparatus for automatically processing an input sequence of data symbols, the apparatus comprising: at least one processor operative to:	To the extent the preamble of claim 12 is found to be limiting, <u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation. <i>See, e.g.</i>, Claim Element [1P], above; <i>see also</i>: Abstract A system and method for creating a database of metadata (metabase) of a variety of digital media content, including TV and radio content delivered on Internet. This semantic-based method captures and enhances domain or subject specific metadata of digital <i>media</i> content, including the specific meaning and intended use of original content. To support semantics, a WorldModel is provided that includes specific domain knowledge, ontologies as well as a set of rules relevant to the original content. The metabase may also be dynamic in that it may track changes to the any variety of accessible content, including live and archival TV and radio programming. FIG. 14, 16:56-17:37:



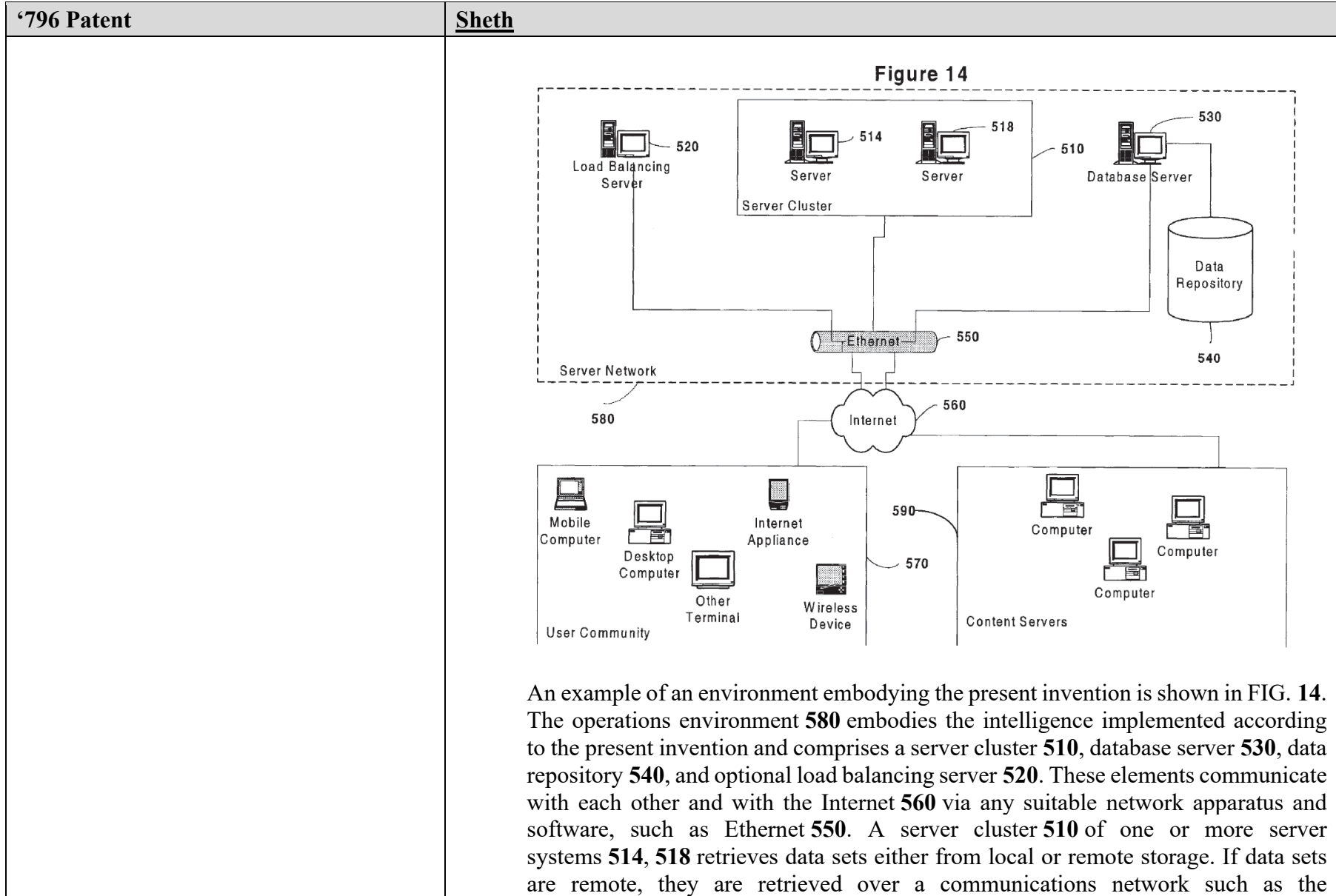
'796 Patent	<u>Sheth</u>
	Internet 560 from content servers 590; however, other communications channels such as dial-up, direction connections, etc. may be used. A data set classifier, such as a URL, is received by the server cluster 510 either before or after retrieval of the data set. The server cluster 510 utilizes the received classifier to retrieve extraction information from a data repository 540. The software intelligence underlying the WorldModel, Metabase Agent, Extractor, WebCrawler and Quality Control Tools are implemented on a database server 530. A user community 570 communicates with the operations environment via the Internet 560. Examples of devices in a user community are mobile computers, desktop computers, terminals, Internet appliances and wireless devices. Although the present invention could be implemented using a variety of programming languages, operating systems or databases, one particular embodiment utilizes the following specific implementation elements. The database used to store the metadata assets is implemented using a database, particularly a relational database. However, other storage architectures such as flat files, hash tables, etc. or database organizations could be used. The software components such as Metabase Agent, Extractor, WebCrawler and Quality Control Tools may be implemented in a particular embodiment utilizing the JAVA programming language. Such programs do not have to meet any stringent performance requirements, but should be highly maintainable, scalable, and platform-independent. Because JAVA is used to implement all of the backend software in this embodiment, extractors may be written, scheduled, and run on both LINUX and WINDOWS® 95/98/NT systems. In the one embodiment, server programs like the Metabase Agent, the relational database, and the extractor scheduling mechanism are run under the LINUX operating system. Other backend tools like the Extractor Toolkit and quality control tools do not require long periods (days) of uptime and can be run on less stable operating systems such as Microsoft WINDOWS®. 17:38-65: In summary, the present invention is directed to a method for generating metadata associated with a data set residing on a storage system. The method comprises steps

'796 Patent	<u>Sheth</u>
	of: receiving a classifier associated with a data set; retrieving WorldModel-based extraction information associated with the received classifier; extracting metadata associated with the data set based upon the retrieved extraction information; and storing in a data store the extracted metadata which is correlated with an identifier corresponding to the data set. Moreover, the present invention is directed to a method for generating targeted content, comprising steps of receiving a classifier associated with a data set; retrieving WorldModel-based extraction information associated with the received classifier; extracting metadata associated with the data set based upon the retrieved extraction information; and accessing content information based on the metadata for presentation to a user. In addition, the present invention is directed to a system for generating metadata associated with a data set, the system comprising: a processor, and a data store coupled to the processor and storing WorldModel-based extraction information. The processor is programmed to retrieve a classifier associated with a data set; retrieve WorldModel-based extraction information associated with a received classifier; extract metadata associated with the data set based upon the retrieved extraction information; and storing in the data store the extracted metadata which is correlated with an identifier corresponding to the data set. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.

'796 Patent	<u>Sheth</u>
[12A] (i) identify at least one regularly identifiable expression in the input sequence of data symbols, wherein the at least one regularly identifiable expression represents a pattern that is matchable in accordance with a programming language that supports such a regularly identifiable expression;	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation for the same reasons as provided above with respect to claim element [1A]. <i>See Claim Element [1A]</i>, above. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
[12B] (ii) identify at least a portion of information associated with the at least one regularly identifiable expression; and	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation for the same reasons as provided above with respect to claim element [1B]. <i>See Claim Element [1B]</i>, above. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
[12C] (iii) extract the portion of information; and	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation for the same reasons as provided above with respect to claim element [1C]. <i>See Claim Element [1C]</i>, above.

EXHIBIT A-Sheth194

'796 Patent	<u>Sheth</u>
	One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
[12D] memory, operatively coupled to the at least one processor, for storing at least a portion of results associated with the identifying and extracting operations.	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation. <i>See, e.g.,</i> Abstract A system and method for creating a database of metadata (metabase) of a variety of digital media content, including TV and radio content delivered on Internet. This semantic-based method captures and enhances domain or subject specific metadata of digital media content, including the specific meaning and intended use of original content. To support semantics, a WorldModel is provided that includes specific domain knowledge, ontologies as well as a set of rules relevant to the original content. The metabase may also be dynamic in that it may track changes to the any variety of accessible content, including live and archival TV and radio programming. FIG. 14, 16:56-17:37:



'796 Patent	<u>Sheth</u>
	Internet 560 from content servers 590; however, other communications channels such as dial-up, direction connections, etc. may be used. A data set classifier, such as a URL, is received by the server cluster 510 either before or after retrieval of the data set. The server cluster 510 utilizes the received classifier to retrieve extraction information from a data repository 540. The software intelligence underlying the WorldModel, Metabase Agent, Extractor, WebCrawler and Quality Control Tools are implemented on a database server 530. A user community 570 communicates with the operations environment via the Internet 560. Examples of devices in a user community are mobile computers, desktop computers, terminals, Internet appliances and wireless devices. Although the present invention could be implemented using a variety of programming languages, operating systems or databases, one particular embodiment utilizes the following specific implementation elements. The database used to store the metadata assets is implemented using a database, particularly a relational database. However, other storage architectures such as flat files, hash tables, etc. or database organizations could be used. The software components such as Metabase Agent, Extractor, WebCrawler and Quality Control Tools may be implemented in a particular embodiment utilizing the JAVA programming language. Such programs do not have to meet any stringent performance requirements, but should be highly maintainable, scalable, and platform-independent. Because JAVA is used to implement all of the backend software in this embodiment, extractors may be written, scheduled, and run on both LINUX and WINDOWS® 95/98/NT systems. In the one embodiment, server programs like the Metabase Agent, the relational database, and the extractor scheduling mechanism are run under the LINUX operating system. Other backend tools like the Extractor Toolkit and quality control tools do not require long periods (days) of uptime and can be run on less stable operating systems such as Microsoft WINDOWS®. 17:38-65: In summary, the present invention is directed to a method for generating metadata associated with a data set residing on a storage system. The method comprises steps

'796 Patent	<u>Sheth</u>
	of: receiving a classifier associated with a data set; retrieving WorldModel-based extraction information associated with the received classifier; extracting metadata associated with the data set based upon the retrieved extraction information; and storing in a data store the extracted metadata which is correlated with an identifier corresponding to the data set. Moreover, the present invention is directed to a method for generating targeted content, comprising steps of receiving a classifier associated with a data set; retrieving WorldModel-based extraction information associated with the received classifier; extracting metadata associated with the data set based upon the retrieved extraction information; and accessing content information based on the metadata for presentation to a user. In addition, the present invention is directed to a system for generating metadata associated with a data set, the system comprising: a processor, and a data store coupled to the processor and storing WorldModel-based extraction information. The processor is programmed to retrieve a classifier associated with a data set; retrieve WorldModel-based extraction information associated with a received classifier; extract metadata associated with the data set based upon the retrieved extraction information; and storing in the data store the extracted metadata which is correlated with an identifier corresponding to the data set. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.

'796 Patent	<u>Sheth</u>
13. The apparatus of claim 12, wherein the regularly identifiable expression identifying operation comprises comparing the input sequence of data symbols to one or more regularly identifiable expressions, previously stored in the memory, to determine if there is a match between a portion of the input sequence and at least one of the previously-stored regularly identifiable expressions.	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation for the same reasons as provided above with respect to claim element [2]. <i>See Claim Element [2]</i>, above. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.

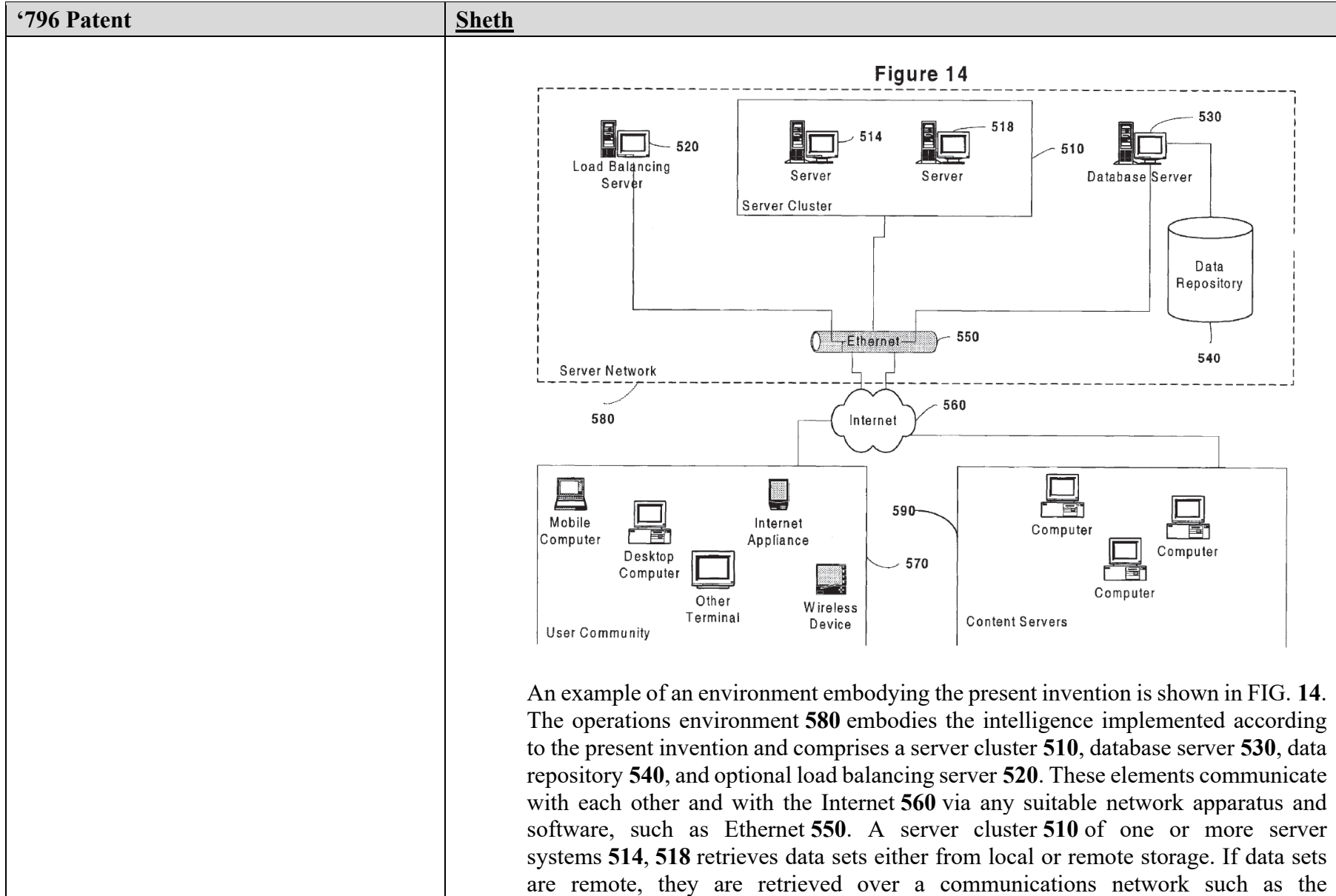
18. The apparatus of claim 12, wherein the extracted portion of information is used to take a specified action.	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation for the same reasons as provided above with respect to claim element [7]. <i>See Claim Element [7]</i>, above.

EXHIBIT A-Sheth194

'796 Patent	<u>Sheth</u>
	One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
21. The apparatus of claim 12, wherein the input sequence of data symbols is representative of at least one of text data, transcribed spoken data, deoxyribonucleic acid sequence data, ribonucleic acid sequence data, amino-acid sequence data, audio sequence data, and video sequence data.	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation for the same reasons as provided above with respect to claim element [10]. <i>See Claim Element [10]</i>, above. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common

'796 Patent	<u>Sheth</u>
	sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.

'796 Patent	<u>Sheth</u>
[23P] An article of manufacture for use in automatically processing an input sequence of data symbols, comprising a machine readable medium containing one or more programs which when executed implement the steps of:	To the extent the preamble of claim 23 is found to be limiting, <u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation. <i>See, e.g., Claim Element [1P]</i>, above; <i>see also</i>: Abstract A system and method for creating a database of metadata (metabase) of a variety of digital media content, including TV and radio content delivered on Internet. This semantic-based method captures and enhances domain or subject specific metadata of digital media content, including the specific meaning and intended use of original content. To support semantics, a WorldModel is provided that includes specific domain knowledge, ontologies as well as a set of rules relevant to the original content. The metabase may also be dynamic in that it may track changes to the any variety of accessible content, including live and archival TV and radio programming. FIG. 14, 16:56-17:37:

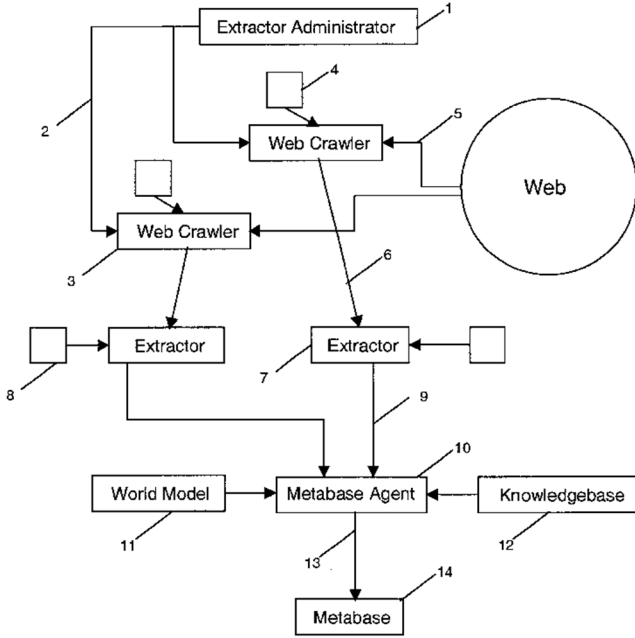


'796 Patent	<u>Sheth</u>
	Internet 560 from content servers 590; however, other communications channels such as dial-up, direction connections, etc. may be used. A data set classifier, such as a URL, is received by the server cluster 510 either before or after retrieval of the data set. The server cluster 510 utilizes the received classifier to retrieve extraction information from a data repository 540. The software intelligence underlying the WorldModel, Metabase Agent, Extractor, WebCrawler and Quality Control Tools are implemented on a database server 530. A user community 570 communicates with the operations environment via the Internet 560. Examples of devices in a user community are mobile computers, desktop computers, terminals, Internet appliances and wireless devices. Although the present invention could be implemented using a variety of programming languages, operating systems or databases, one particular embodiment utilizes the following specific implementation elements. The database used to store the metadata assets is implemented using a database, particularly a relational database. However, other storage architectures such as flat files, hash tables, etc. or database organizations could be used. The software components such as Metabase Agent, Extractor, WebCrawler and Quality Control Tools may be implemented in a particular embodiment utilizing the JAVA programming language. Such programs do not have to meet any stringent performance requirements, but should be highly maintainable, scalable, and platform-independent. Because JAVA is used to implement all of the backend software in this embodiment, extractors may be written, scheduled, and run on both LINUX and WINDOWS® 95/98/NT systems. In the one embodiment, server programs like the Metabase Agent, the relational database, and the extractor scheduling mechanism are run under the LINUX operating system. Other backend tools like the Extractor Toolkit and quality control tools do not require long periods (days) of uptime and can be run on less stable operating systems such as Microsoft WINDOWS®. 17:38-65: In summary, the present invention is directed to a method for generating metadata associated with a data set residing on a storage system. The method comprises steps

'796 Patent	<u>Sheth</u>
	of: receiving a classifier associated with a data set; retrieving WorldModel-based extraction information associated with the received classifier; extracting metadata associated with the data set based upon the retrieved extraction information; and storing in a data store the extracted metadata which is correlated with an identifier corresponding to the data set. Moreover, the present invention is directed to a method for generating targeted content, comprising steps of receiving a classifier associated with a data set; retrieving WorldModel-based extraction information associated with the received classifier; extracting metadata associated with the data set based upon the retrieved extraction information; and accessing content information based on the metadata for presentation to a user. In addition, the present invention is directed to a system for generating metadata associated with a data set, the system comprising: a processor, and a data store coupled to the processor and storing WorldModel-based extraction information. The processor is programmed to retrieve a classifier associated with a data set; retrieve WorldModel-based extraction information associated with a received classifier; extract metadata associated with the data set based upon the retrieved extraction information; and storing in the data store the extracted metadata which is correlated with an identifier corresponding to the data set. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.

'796 Patent	<u>Sheth</u>
[23A] identifying at least one regularly identifiable expression in the input sequence of data symbols, wherein the at least one regularly identifiable expression represents a pattern that is matchable in accordance with a programming language that supports such a regularly identifiable expression;	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation for the same reasons as provided above with respect to claim element [1A]. <i>See Claim Element [1A]</i>, above. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
[23B] identifying at least a portion of information associated with the at least one regularly identifiable expression; and	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation for the same reasons as provided above with respect to claim element [1B]. <i>See Claim Element [1B]</i>, above. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
[23C] extracting the portion of information.	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation for the same reasons as provided above with respect to claim element [1C]. <i>See Claim Element [1C]</i>, above.

'796 Patent	<u>Sheth</u>
	One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
[24P] Apparatus for automatically processing an input sequence of data symbols, the apparatus comprising:	To the extent the preamble of claim 24 is found to be limiting, <u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation for the same reasons as provided above with respect to claim element [12P]. <i>See Claim Element [12P]</i>, above. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
[24A] a data capture device for obtaining the input sequence of data symbols;	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation. <i>See, e.g.</i>, FIG. 5, 6:6-7, 10:7-67, 11:11-22, 11:37-55, 12:34-13:2:

'796 Patent	Sheth
	 The diagram illustrates the extraction process. At the top, the 'Extractor Administrator' (1) manages the system. Two 'Web Crawlers' (3) are shown, one of which is controlled by a small square icon (4). The 'Web Crawlers' (3) interact with the 'Web' (represented by a circle) and receive input from a small square icon (5). They feed data into two 'Extractor' modules (7). The 'Extractors' (7) also receive input from small square icons (8) and (9). The extracted data is then processed by the 'Metabase Agent' (10), which also receives input from the 'World Model' (11) and the 'Knowledgebase' (12). The 'Metabase Agent' (10) then stores the data in the 'Metabase' (14) via a connection (13). Fig. 5 FIG. 5 shows the extraction process during which metadata for digital media is discovered, copied from a Web site, enhanced, and inserted into the present invention's database. Five modules contribute to this task: 1) WebCrawlers 3 that scan certain Web sites; 2) Extractors 7 that do the actual metadata extraction; 3) Knowledgebase 12 that improves the metadata where possible; 4) Metabase Agent 10 that inserts the data into the present invention's metabase; and 5) Extractor Administrator 1 that schedules the running of WebCrawlers.

'796 Patent	<u>Sheth</u>
	The Extractor Administration 1 module maintains a list of all available crawlers/extractors along with a description of when the respective extraction process should be started and initiates extraction according to the schedule. The process starts by identifying a source containing digital media data sets. Several factors determine whether a source is actually worth extracting. The foregoing discussion refers to a Web site as an example source of digital media for retrieval; other potential sources such as a local drive or another remote resource such as FTP, GOPHER, etc. type material will be understood by those of skill in the art to be encompassed within the scope of the present invention. These include the volume of media files and the quality of the available metadata about those media files. Once a Web site is deemed worthy of undergoing this process, the assets that will be acquired from the site are assigned to a particular domain in the WorldModel. Next, crawling and extraction rules are written that specify where and how to retrieve the metadata from the Web site. For each attribute of a particular asset type (domain), an extractor writer provides a rule that specifies where to find a value of this attribute. Crawling and extraction rules are input to WebCrawler and Extractor programs that traverse Web sites and retrieve digital media metadata from selected pages. An extractor program takes HTML pages and extraction rules as input and generates XML assets such as that shown in FIG. 6. These generated assets contain values for each attribute name belonging to the domain of that Web site. Once created, the assets are sent to a Metabase Agent that is in charge of enhancing and inserting them into a database of records. In order to enhance the assets, the Metabase Agent uses information stored in the WorldModel as well as a Knowledgebase. The Knowledgebase is a collection of tables containing domain-specific information and relationships. After insertion into the metabase, the assets are then ready to be searched. The purpose of a WebCrawler is to read in the content of “extractable” Web pages from a site and pass this content on to the Extractors. A WebCrawler 3 is a piece of software, invoked on a remote or local host, which begins reading pages from a particular site and determines which of these pages are extractable. When a WebCrawler is invoked 2, it is given a list of “crawling rules” 4 specific to a single

'796 Patent	<u>Sheth</u>
	Web site. These rules dictate where (on which page) the crawler should begin its search, which directories the crawler must remain within, and define the characteristics of an extractable page. Without such rules, a WebCrawler would likely find a link off of the site it was assigned to crawl and begin aimlessly reading the entire Web. Extractors 7 are programs that are designed to find information about digital media from a Web page. Their ability to retrieve values for domain-specific attributes is critical to capturing the semantics of the information on the source site. Extractors exploit Web pages that follow a template or pages that are generated by server-side scripts/programs. By assuming that Web pages from a particular site follow a pattern or have some recognizable structure, rules can be written that reliably retrieve small sections of text from those pages. ... When a WebCrawler finds an extractable page it will send this page (as well as the name of the site on which it is crawling) to one extractor in a large pool of extractors running on another machine. An extractor is designed to work on only one of these pages a time. Once an extractor receives a page, it looks up a set of extraction rules 8 associated with the site from which the page came. These rules list the metadata attributes for the type of media that this site contains as well as rules that describe where to find values for these attributes within the page. The set of attributes associated with, for example, a news video (reporter, location, event date, etc.) is different from the set of attributes associated with a sports highlight (teams, players, score etc.) The extractor scans the Web page content for pieces of text that match the pattern specified by the extraction rules. After the extractor has attempted to find every attribute in its list of extraction rules, it creates an XML document containing attribute-value pairs 9. This document is sent to the Metabase Agent 10. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this

'796 Patent	<u>Sheth</u>
	reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
[24B] at least one processor, operatively coupled to the data capture device, operative to: (i) identify at least one regularly identifiable expression in the input sequence of data symbols. wherein the at least one regularly identifiable expression represents a pattern that is matchable in accordance with a programming language that supports such a regularly identifiable expression; (ii) identify at least a portion of information associated with the at least one regular expression; and (iii) extract the portion of information;	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation for the same reasons as provided above with respect to claim element [12A]. <i>See Claim Element [12A]</i>, above. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
[24C] memory, operatively coupled to the at least one processor, for storing at least a portion of results associated with the identifying and extracting operations; and	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation for the same reasons as provided above with respect to claim element [12B]. <i>See Claim Element [12B]</i>, above. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other

'796 Patent	<u>Sheth</u>
	prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
[24D] a data output device, operatively coupled to the at least one processor, for presenting the extracted portion of information to a user.	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation for the same reasons as provided above with respect to claim element [8]. <i>See Claim Element [8]</i>, above. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
[25P] A method of automatically processing an input document, the method comprising the steps of:	To the extent the preamble of claim 25 is found to be limiting, <u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation for the same reasons as provided above with respect to claim element [1P]. <i>See Claim Element [1P]</i>, above. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
[25A] identifying one or more regular expressions in the input document by	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation for the same reasons

'796 Patent	<u>Sheth</u>
comparing the input document to one or more previously-stored regular expressions to determine if there is a match between a portion of the input document and at least one of the previously-stored regular expressions, wherein a regular expression represents a pattern that is matchable in accordance with a programming language that supports such a regular expression; and	as provided above with respect to claim element [1A] and [2]. <i>See Claim Elements [1A] and [2]</i>, above. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
[25B] identifying at least a portion of information associated with the at least one regular expression for extraction.	<u>Sheth</u> discloses this claim limitation explicitly, inherently, or as a matter of common sense, or it would have been obvious to add missing aspects of the limitation for the same reasons as provided above with respect to claim elements [1B] and [1C]. <i>See Claim Elements [1B]-[1C]</i>, above. One of ordinary skill would find this limitation disclosed either expressly or inherently in the teachings of this reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention. To the extent this reference is not found to teach this element explicitly, implicitly, or inherently, the element would have been obvious to one of ordinary skill in the art based on this reference, common sense, ordinary creativity of one of ordinary skill in the art, and the state of the art. Additionally, it would have been obvious to combine this reference with one or more other prior art references identified in Defendant's Invalidity Contentions Cover Pleading. Rather than repeat those disclosures here, they are incorporated by reference into this chart.