TurboPatent™

Expert patent professionals + advanced AI
equals more affordable, better quality patents

TurboPatent Sample Patent

PERSONAL INFORMATION DATA MANAGER

US 9,619,661

April 11, 2017

US009619661B1

(12) **United States Patent**
Finkelstein

(10) **Patent No.: US 9,619,661 B1**
(45) **Date of Patent: Apr. 11, 2017**

(54) **PERSONAL INFORMATION DATA MANAGER**

(71) Applicant: **Charles Finkelstein Consulting LLC**, Woodinville, WA (US)

(72) Inventor: **Charles Finkelstein**, Woodinville, WA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **14/742,655**

(22) Filed: **Jun. 17, 2015**

Related U.S. Application Data

(60) Provisional application No. 62/013,250, filed on Jun. 17, 2014.

(51) **Int. Cl.**
G06F 7/04 (2006.01)
G06F 15/16 (2006.01)
G06F 17/30 (2006.01)
H04L 29/06 (2006.01)
G06F 21/60 (2013.01)

(52) **U.S. Cl.**
CPC ***G06F 21/604*** (2013.01)

(58) **Field of Classification Search**
CPC ... H04L 63/08; H04L 63/083; H04L 63/0861; H04W 12/06; G06F 21/31
USPC 726/7
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

2009/0287837 A1* 11/2009 Felsher G06F 19/322 709/229
2010/0317420 A1* 12/2010 Hoffberg G06Q 30/0207 463/1
2013/0159021 A1* 6/2013 Felsher G06F 19/322 705/3

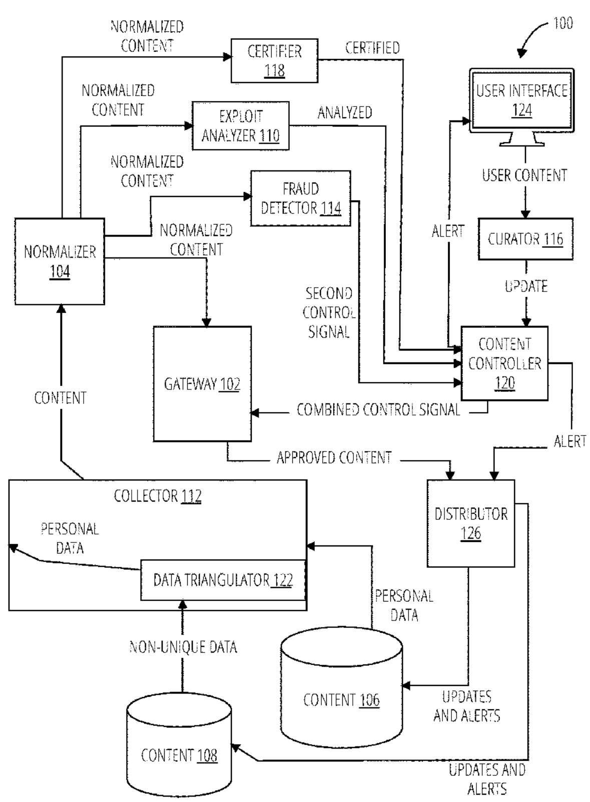
* cited by examiner

Primary Examiner — Kambiz Zand
Assistant Examiner — Stephen Sanders
(74) *Attorney, Agent, or Firm* — FSP LLC

(57) **ABSTRACT**

A method and system for managing personal data is provided as a means to increase the efficiency and effectiveness of personal data access, approval and curation across multiple content sources. The method and system accomplishes this by discovering, identifying, collecting normalizing personal information from content sources then alerting the user to any identifiable use of their personal information and allowing the user to directly dictate personal information use policies and manually control individual personal data attributes on multiple content platforms.

5 Claims, 7 Drawing Sheets



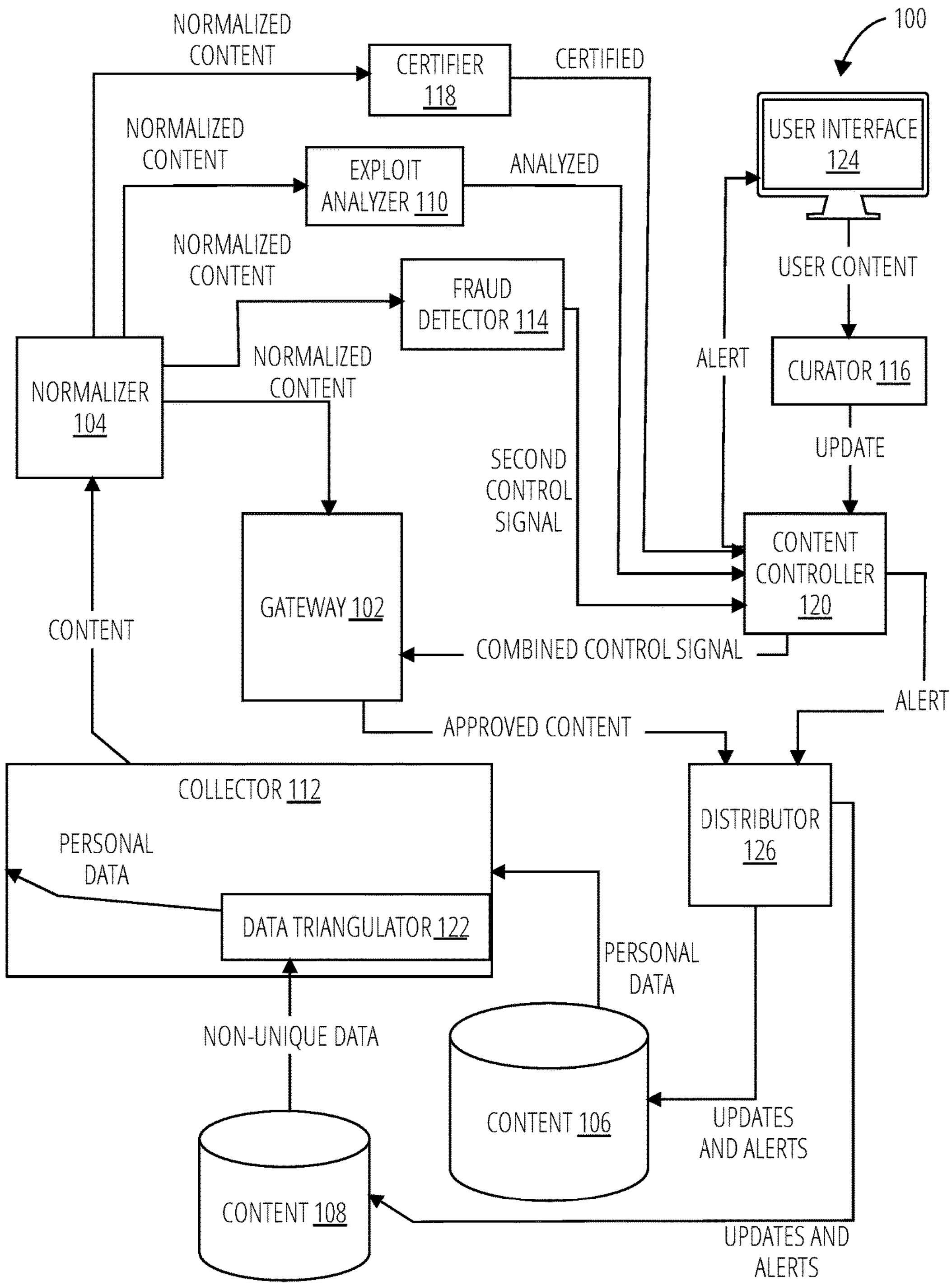

100
NORMALIZED CONTENT
CERTIFIER 118
CERTIFIED
USER INTERFACE 124
NORMALIZED CONTENT
EXPLOIT ANALYZER 110
ANALYZED
NORMALIZED CONTENT
FRAUD DETECTOR 114
USER CONTENT
ALERT
CURATOR 116
NORMALIZER 104
NORMALIZED CONTENT
UPDATE
SECOND CONTROL SIGNAL
GATEWAY 102
CONTENT CONTROLLER 120
CONTENT
COMBINED CONTROL SIGNAL
ALERT
APPROVED CONTENT
COLLECTOR 112
DISTRIBUTOR 126
PERSONAL DATA
DATA TRIANGULATOR 122
PERSONAL DATA
NON-UNIQUE DATA
CONTENT 106
UPDATES AND ALERTS
CONTENT 108
UPDATES AND ALERTS

FIG. 1

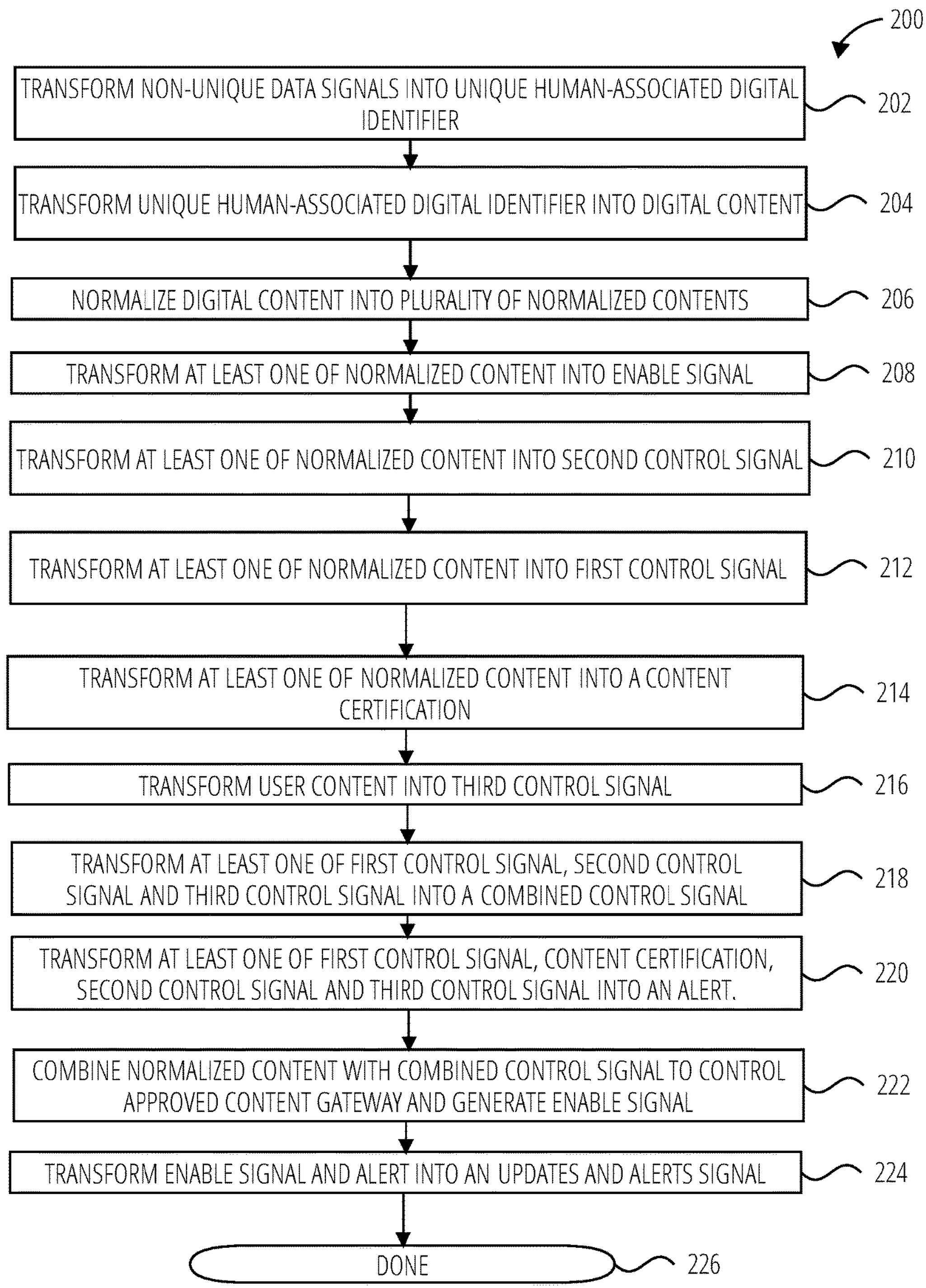

200
TRANSFORM NON-UNIQUE DATA SIGNALS INTO UNIQUE HUMAN-ASSOCIATED DIGITAL IDENTIFIER
202
TRANSFORM UNIQUE HUMAN-ASSOCIATED DIGITAL IDENTIFIER INTO DIGITAL CONTENT
204
NORMALIZE DIGITAL CONTENT INTO PLURALITY OF NORMALIZED CONTENTS
206
TRANSFORM AT LEAST ONE OF NORMALIZED CONTENT INTO ENABLE SIGNAL
208
TRANSFORM AT LEAST ONE OF NORMALIZED CONTENT INTO SECOND CONTROL SIGNAL
210
TRANSFORM AT LEAST ONE OF NORMALIZED CONTENT INTO FIRST CONTROL SIGNAL
212
TRANSFORM AT LEAST ONE OF NORMALIZED CONTENT INTO A CONTENT CERTIFICATION
214
TRANSFORM USER CONTENT INTO THIRD CONTROL SIGNAL
216
TRANSFORM AT LEAST ONE OF FIRST CONTROL SIGNAL, SECOND CONTROL SIGNAL AND THIRD CONTROL SIGNAL INTO A COMBINED CONTROL SIGNAL
218
TRANSFORM AT LEAST ONE OF FIRST CONTROL SIGNAL, CONTENT CERTIFICATION, SECOND CONTROL SIGNAL AND THIRD CONTROL SIGNAL INTO AN ALERT.
220
COMBINE NORMALIZED CONTENT WITH COMBINED CONTROL SIGNAL TO CONTROL APPROVED CONTENT GATEWAY AND GENERATE ENABLE SIGNAL
222
TRANSFORM ENABLE SIGNAL AND ALERT INTO AN UPDATES AND ALERTS SIGNAL
224
DONE
226

FIG. 2

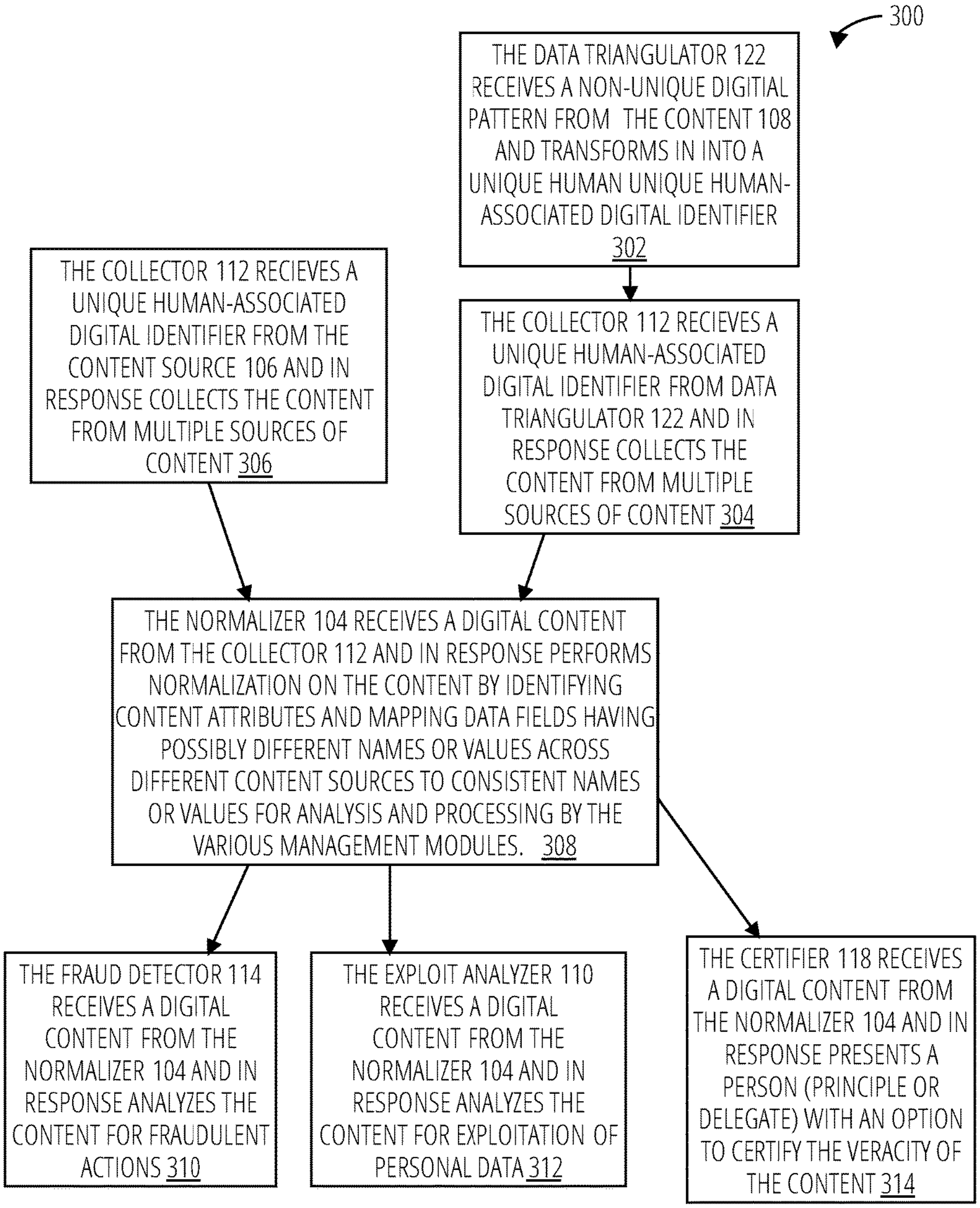

300
THE DATA TRIANGULATOR 122 RECEIVES A NON-UNIQUE DIGITIAL PATTERN FROM THE CONTENT 108 AND TRANSFORMS IN INTO A UNIQUE HUMAN UNIQUE HUMAN-ASSOCIATED DIGITAL IDENTIFIER 302
THE COLLECTOR 112 RECIEVES A UNIQUE HUMAN-ASSOCIATED DIGITAL IDENTIFIER FROM DATA TRIANGULATOR 122 AND IN RESPONSE COLLECTS THE CONTENT FROM MULTIPLE SOURCES OF CONTENT 304
THE COLLECTOR 112 RECIEVES A UNIQUE HUMAN-ASSOCIATED DIGITAL IDENTIFIER FROM THE CONTENT SOURCE 106 AND IN RESPONSE COLLECTS THE CONTENT FROM MULTIPLE SOURCES OF CONTENT 306
THE NORMALIZER 104 RECEIVES A DIGITAL CONTENT FROM THE COLLECTOR 112 AND IN RESPONSE PERFORMS NORMALIZATION ON THE CONTENT BY IDENTIFYING CONTENT ATTRIBUTES AND MAPPING DATA FIELDS HAVING POSSIBLY DIFFERENT NAMES OR VALUES ACROSS DIFFERENT CONTENT SOURCES TO CONSISTENT NAMES OR VALUES FOR ANALYSIS AND PROCESSING BY THE VARIOUS MANAGEMENT MODULES. 308
THE FRAUD DETECTOR 114 RECEIVES A DIGITAL CONTENT FROM THE NORMALIZER 104 AND IN RESPONSE ANALYZES THE CONTENT FOR FRAUDULENT ACTIONS 310
THE EXPLOIT ANALYZER 110 RECEIVES A DIGITAL CONTENT FROM THE NORMALIZER 104 AND IN RESPONSE ANALYZES THE CONTENT FOR EXPLOITATION OF PERSONAL DATA 312
THE CERTIFIER 118 RECEIVES A DIGITAL CONTENT FROM THE NORMALIZER 104 AND IN RESPONSE PRESENTS A PERSON (PRINCIPLE OR DELEGATE) WITH AN OPTION TO CERTIFY THE VERACITY OF THE CONTENT 314

FIG. 3

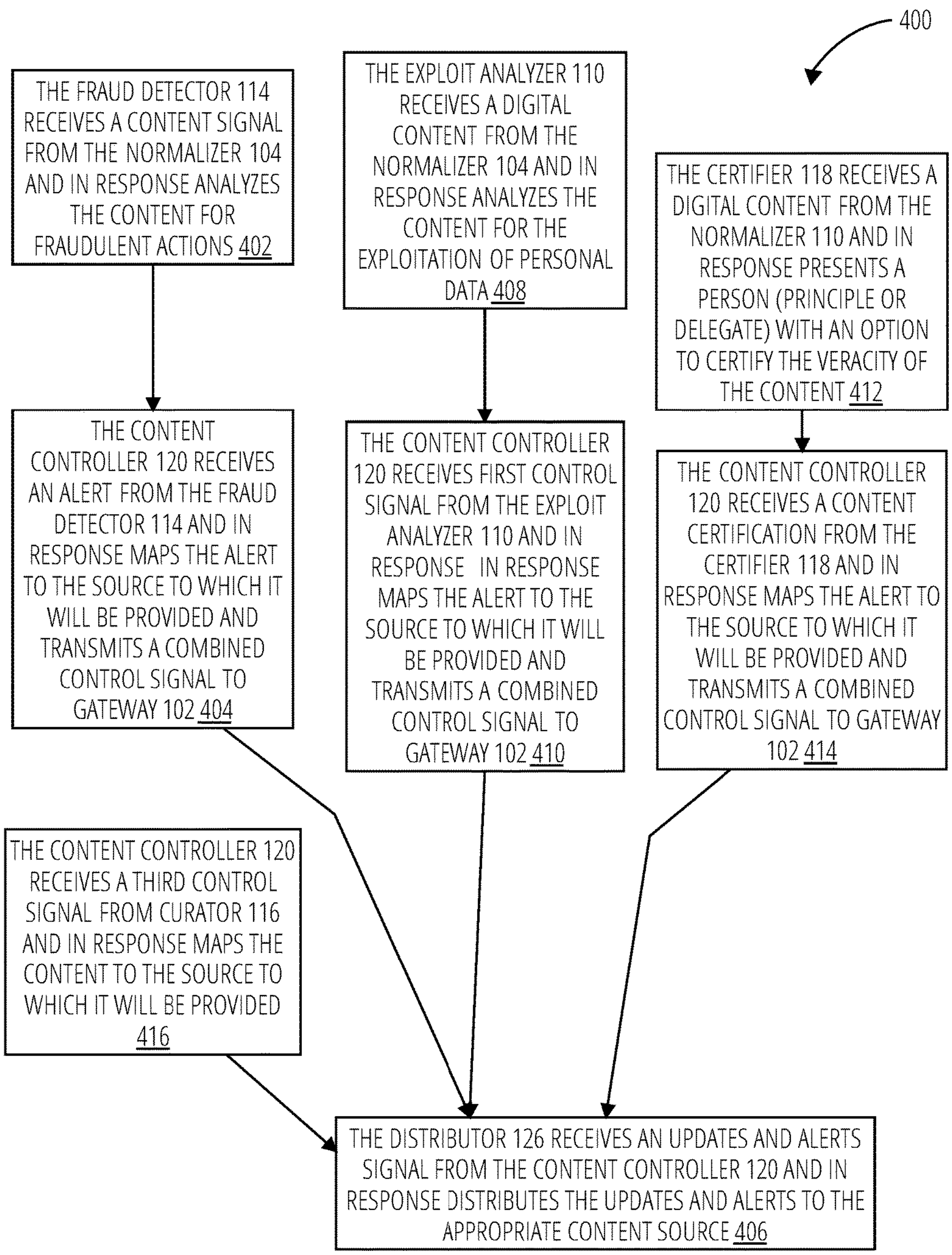

400
THE FRAUD DETECTOR 114 RECEIVES A CONTENT SIGNAL FROM THE NORMALIZER 104 AND IN RESPONSE ANALYZES THE CONTENT FOR FRAUDULENT ACTIONS 402
THE EXPLOIT ANALYZER 110 RECEIVES A DIGITAL CONTENT FROM THE NORMALIZER 104 AND IN RESPONSE ANALYZES THE CONTENT FOR THE EXPLOITATION OF PERSONAL DATA 408
THE CERTIFIER 118 RECEIVES A DIGITAL CONTENT FROM THE NORMALIZER 110 AND IN RESPONSE PRESENTS A PERSON (PRINCIPLE OR DELEGATE) WITH AN OPTION TO CERTIFY THE VERACITY OF THE CONTENT 412
THE CONTENT CONTROLLER 120 RECEIVES AN ALERT FROM THE FRAUD DETECTOR 114 AND IN RESPONSE MAPS THE ALERT TO THE SOURCE TO WHICH IT WILL BE PROVIDED AND TRANSMITS A COMBINED CONTROL SIGNAL TO GATEWAY 102 404
THE CONTENT CONTROLLER 120 RECEIVES FIRST CONTROL SIGNAL FROM THE EXPLOIT ANALYZER 110 AND IN RESPONSE IN RESPONSE MAPS THE ALERT TO THE SOURCE TO WHICH IT WILL BE PROVIDED AND TRANSMITS A COMBINED CONTROL SIGNAL TO GATEWAY 102 410
THE CONTENT CONTROLLER 120 RECEIVES A CONTENT CERTIFICATION FROM THE CERTIFIER 118 AND IN RESPONSE MAPS THE ALERT TO THE SOURCE TO WHICH IT WILL BE PROVIDED AND TRANSMITS A COMBINED CONTROL SIGNAL TO GATEWAY 102 414
THE CONTENT CONTROLLER 120 RECEIVES A THIRD CONTROL SIGNAL FROM CURATOR 116 AND IN RESPONSE MAPS THE CONTENT TO THE SOURCE TO WHICH IT WILL BE PROVIDED 416
THE DISTRIBUTOR 126 RECEIVES AN UPDATES AND ALERTS SIGNAL FROM THE CONTENT CONTROLLER 120 AND IN RESPONSE DISTRIBUTES THE UPDATES AND ALERTS TO THE APPROPRIATE CONTENT SOURCE 406

FIG. 4

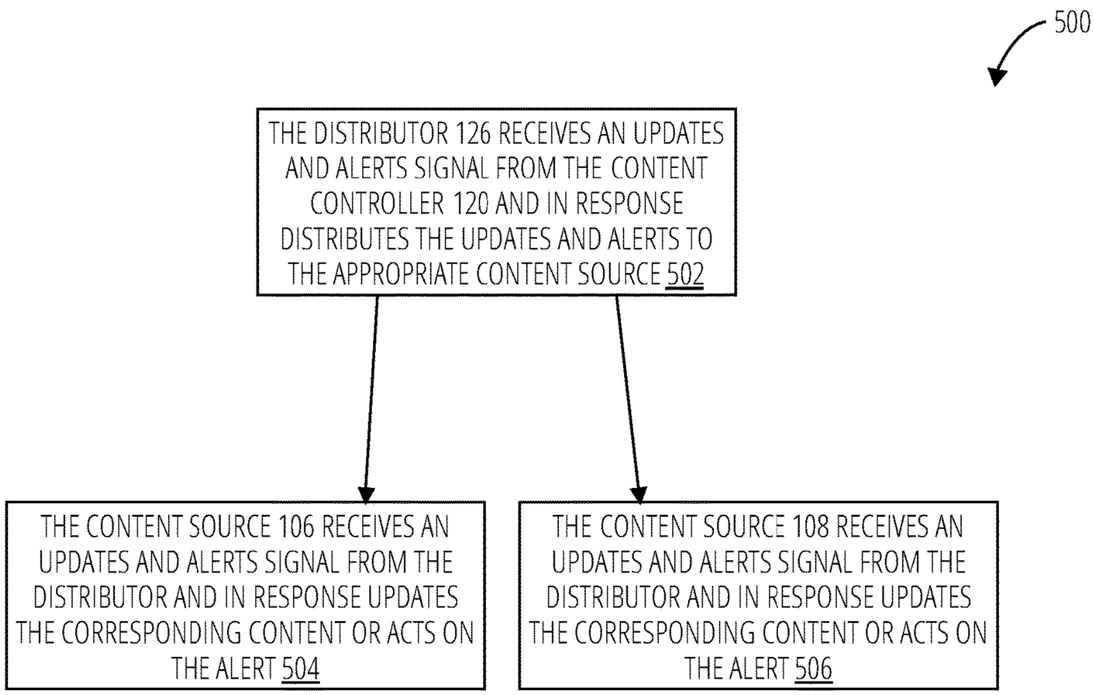

500
THE DISTRIBUTOR 126 RECEIVES AN UPDATES AND ALERTS SIGNAL FROM THE CONTENT CONTROLLER 120 AND IN RESPONSE DISTRIBUTES THE UPDATES AND ALERTS TO THE APPROPRIATE CONTENT SOURCE 502
THE CONTENT SOURCE 106 RECEIVES AN UPDATES AND ALERTS SIGNAL FROM THE DISTRIBUTOR AND IN RESPONSE UPDATES THE CORRESPONDING CONTENT OR ACTS ON THE ALERT 504
THE CONTENT SOURCE 108 RECEIVES AN UPDATES AND ALERTS SIGNAL FROM THE DISTRIBUTOR AND IN RESPONSE UPDATES THE CORRESPONDING CONTENT OR ACTS ON THE ALERT 506

FIG. 5

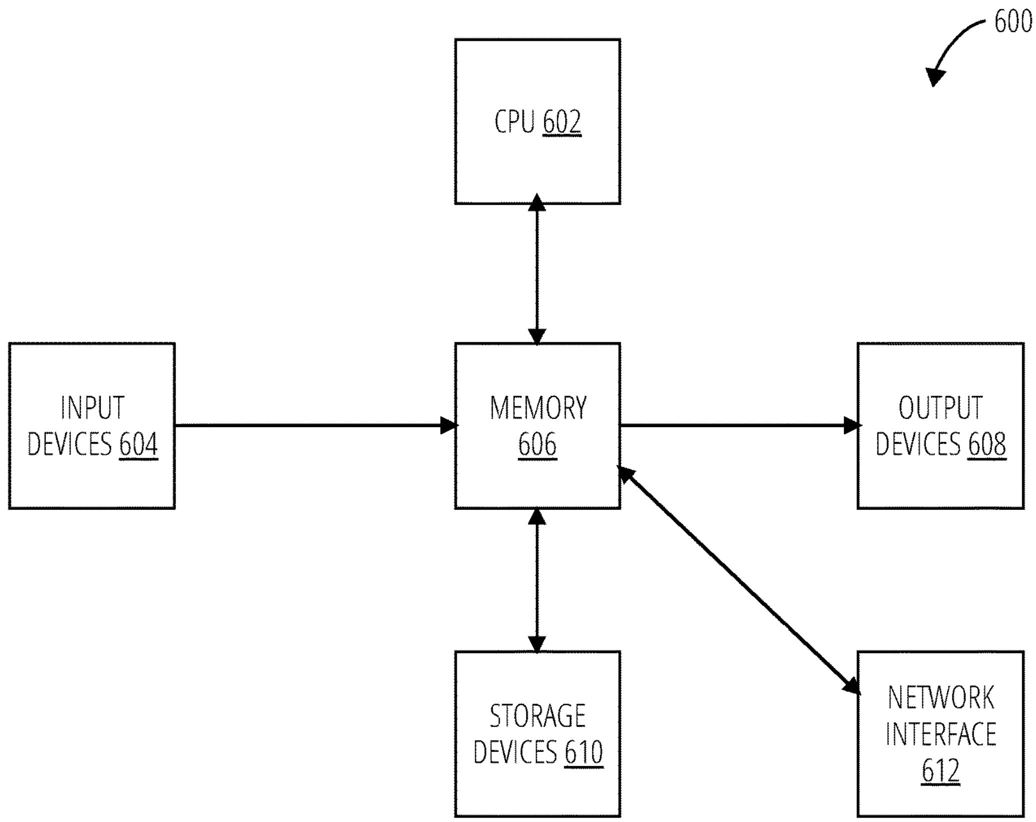
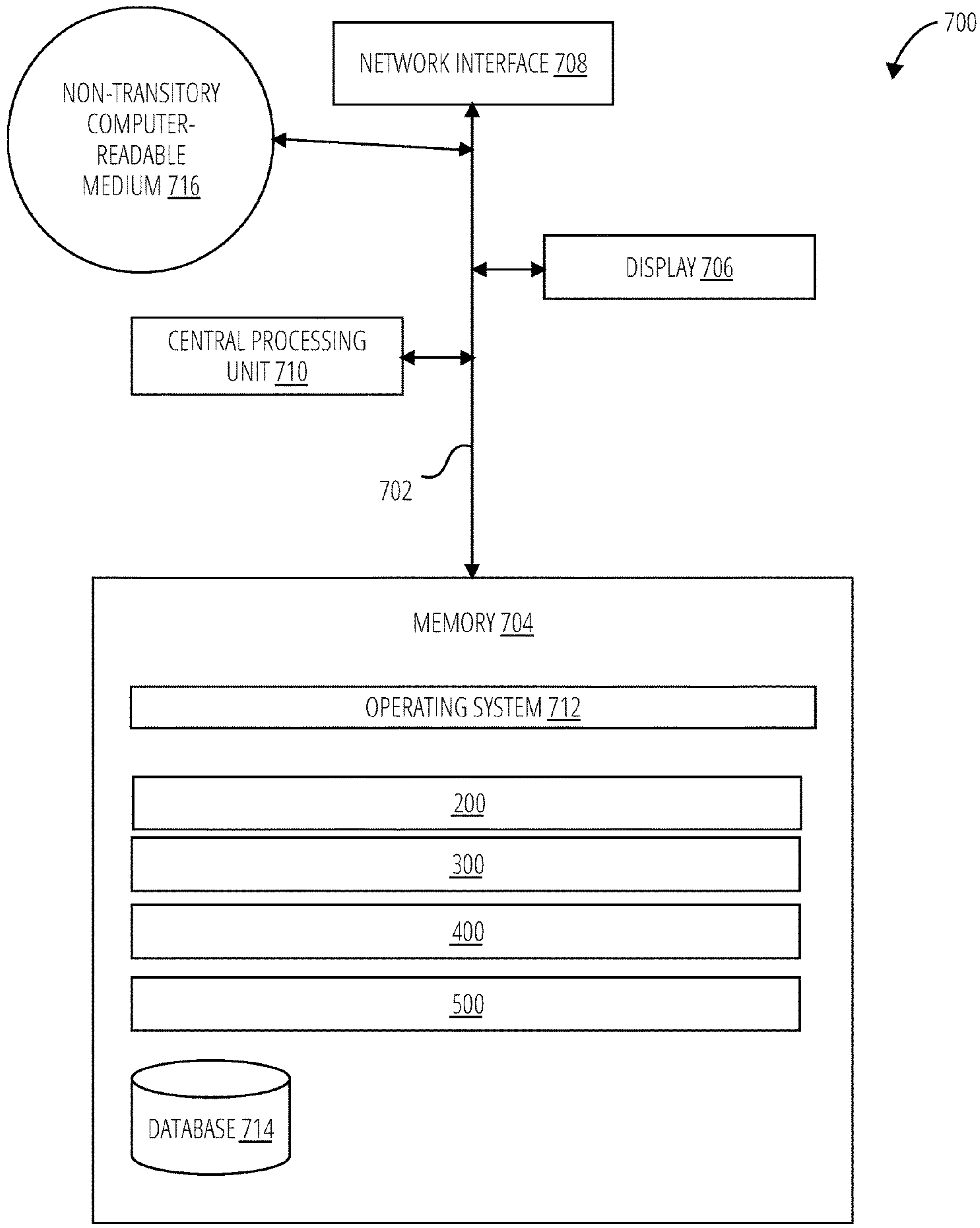

600
CPU 602
INPUT DEVICES 604
MEMORY 606
OUTPUT DEVICES 608
STORAGE DEVICES 610
NETWORK INTERFACE 612

FIG. 6



700
NON-TRANSITORY COMPUTER-READABLE MEDIUM 716
NETWORK INTERFACE 708
DISPLAY 706
CENTRAL PROCESSING UNIT 710
702
MEMORY 704
OPERATING SYSTEM 712
200
300
400
500
DATABASE 714

FIG. 7

PERSONAL INFORMATION DATA MANAGER

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority and benefit under 35 U.S.C. 119 to U.S. application Ser. No. 62/013,250, filed on Jun. 17, 2014, which is incorporated by reference herein in its entirety.

BACKGROUND

With the proliferation internet technology, and the development of big data and social networking, personal data is becoming ever-increasingly public. It is also becoming increasingly difficult to update personal information across manage what entities have access to your personal data, whether or not these entities are showing accurate information, and whether or not they are improperly using said data.

BRIEF SUMMARY

The following summary is intended to highlight and introduce some aspects of the disclosed embodiments, but not to limit the scope of the claims. Thereafter, a detailed description of illustrated embodiments is presented, which will permit one skilled in the relevant art to make and use various embodiments.

Embodiments of a system and methods are described which discovers, tracks and manages the release and update of personal information throughout it's lifecycle across multiple applications and web-based environments.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

To easily identify the discussion of any particular element or act, the most significant digit or digits in a reference number refer to the figure number in which that element is first introduced.

FIG. **1** is a system diagram of an embodiment of a personal data management system.

FIG. **2** illustrates a routine for managing personal data in accordance with one embodiment.

FIG. **3** illustrates a routine for managing personal data in accordance with one embodiment.

FIG. **4** illustrates a routine for managing personal data in accordance with one embodiment.

FIG. **5** illustrates a routine for managing personal data in accordance with one embodiment.

FIG. **6** illustrates an embodiment of a digital apparatus **600** to implement components and process steps of the system described herein.

FIG. **7** illustrates a server **700** in accordance with one embodiment.

DETAILED DESCRIPTION

Glossary

"content source" in this context refers to a website, application or other internet-based location which contains personal data

"security question answers" in this context refers to answers given to a user authentication question such as "What is your mother's maiden name?"

"non-unique personal data attributes" in this context refers to such attributes that are not unique to an individual, such as height, name, weight, sexual orientation.

"unique personal data attributes" in this context refers to such attributes that are unique to an individual, such as a social security number.

DESCRIPTION

The phrases "in one embodiment", "in various embodiments", "in some embodiments", and the like are used repeatedly. Such phrases do not necessarily refer to the same embodiment. The terms "comprising", "having", and "including" are synonymous, unless the context dictates otherwise.

Reference is now made in detail to the description of the embodiments as illustrated in the drawings. While embodiments are described in connection with the drawings and related descriptions, there is no intent to limit the scope to the embodiments disclosed herein. On the contrary, the intent is to cover all alternatives, modifications and equivalents. In alternate embodiments, additional devices, or combinations of illustrated devices, may be added to or combined, without limiting the scope to the embodiments disclosed herein.

A system embodiment is described herein that includes components and logic to implement a universal client having a user interface to visualize complex personal data distributed across many content sources on the Internet (or other wide area networks). The data is visualized by the system in manners relevant to the person, who is enabled to control relevance/context scoring vectors for individual or logical groups of personal data. The system also includes a universal data connector service for collecting/crawling personal data, a data curation service accessible from the user interface, a distributor service for exporting/pushing content back out to content sources, and a synchronization services that is notified when external sources have updates, imports, and any changes are exported and "sync'd' back to the content sources. The system is useful for managing personal data for consumers (private persons), businesses (entities that monetize personal information), enterprise customers (knowledge workers within organizations), as well as business-to-business.

In some embodiments, a system may include a data triangulator to transform a non-unique digital pattern into a unique human-associated digital identifier; a collector to transform the unique human-associated digital identifier into digital content; a normalizer to transform the digital content into a group of normalized content; a certifier to transform the normalized content into a content certification; an exploit analyzer to transform the normalized content into a first control signal; a fraud detector to transform the normalized content into a second control signal; a curator to transform a user content into a third control signal; a content controller to transform the third control signal, the content certification, the first control signal, and the second control signal into an alert and a combined control signal; a gateway to transform the combined control signal and the normalized content into an enable signal; and/or a distributor to respond to the enable signal and the alert to communicate the normalized content to at least one machine server.

In some embodiments, the normalizer may include logic to receive the digital content and convert the digital content into the normalized content with standardized constituent personal data attributes, logic to send the digital content to the certifier, logic to send the digital content to the exploit analyzer, and/or logic to send the digital content to the fraud detector.

In some embodiments, the exploit analyzer may include Logic to apply a user data policy containing rules for data usage to the normalized content to create the first control signal.

In some embodiments, the fraud detector may include logic to compare the normalized content to a group of user data attributes such as social security number, current address, previous address, date of birth, mother's maiden name, and security question answers.

In some embodiments, the collector may include logic to identify personal information from a content source and/or logic to transmit collected content to the normalizer logic.

In some embodiments, the data triangulator may include logic to parse the non-unique digital pattern into at least one non-unique personal data attributes, logic to compare the non-unique personal data attributes to user personal data attributes and compile into the unique human-associated digital identifier, and/or logic to transmit the unique human-associated digital identifier to the collector.

In some embodiments, the content controller may include logic to receive at least one of the content certification, the first control signal, and the second control signal and generate the alert corresponding to the received signal and/or logic to receive at least one of the content certification, the first control signal, the second control signal and the third control signal, and combine the signals into the combined control signal.

In some embodiments, the data triangulator may include logic to aggregate the non-unique personal data attributes from content signals and determine the similarity of the non-unique personal data attributes with the non-unique personal data attributes with the user's non-unique personal data attributes.

1. Viewing and curating data on subscription based IaaS applications such as Facebook, Amazon, Microsoft, Google, or viewing and curating data on subscription based B 2 B data brokers such as Axciom or marketing data providers such as Datalogix, comscore, Gatner and others or viewing and curating data on subscription based B 2 C data REPORTING agencies such as Equifax and TransUnion, or bacground information aggregators/providers Intelius or viewing and curating personal data stored with retailers such Target and Walmart,

a. User logs into Personal data App (with 2nd factor authentication)

b. The application shows in tree view and other views the Web based services that have their personal data.

c. The personal data app has been authenticated by standard means to access their current data in these systems, and/or may do common screen scraping techniques—these actions happen in the background as a service and/or when the user hits update refresh on selected services

d. For example when the user decided to view Facebook, they see the data the Facebook explicitly has from the user:

i. Basic PII

1. Legal name

2. Social security number (if available)

3. Title

4. Gender

5. Height

6. Weight

7. Ethnicity

8. Age

9. Address(s)

10. Phone number(s)

11. Email address(es)

12. Other contact info, (e.g. Web page, twitter, etc)

ii. Extended PII

1. Marital status

2. Education history, grades and status

3. Veteran status

4. Sexual preference(s)

5. Credit score

6. Relationship preferences

7. Income if available

8. Place(s) of work

9. Places where they live(d)

10. Family and relationships

11. Life events (e.g. weddings, births, graduations, vacations, incl place, date, time)

12. Hobbies

13. Interests

14. Health conditions

15. Memberships and affiliations

16. Political classifications and affiliations

iii. Consumption PII (white list and black list)

1. Music and Entertainment preferences (white list and black list)

2. Food interests

3. Health interests

4. Lifestyle interests

5. Recreational interest (e.g. bike, hike ski.

6. Vacation interests

7. Informational preferences (news, books, etc.)

8. Brand preferences in all categories

9. Influencers

10. Retailer preferences

11. Medical preferences

12. Medical procedures

13. Credit score

14. Insurance ratings

15. Accident histories

16. Everything else

e. The user views these details—each one has a date from created, last modified, etc, and opt-in/legal note for in-policy or out of policy (controlled by the user, and meeting local laws)

f. They may edit, add or delete any of them. The associated meta data in e is updated.

g. They then push the updates back out to the services—

i. where they have direct access, the information is auto updated,

ii. where they do not have access to change, it is in the services update queue—must comply to local laws

h. The user may then see a log of the updates and then view them

i. the user may do them for each services a-h for each service individually or perform global changes, via our normalization algorithms

j. The user may run "monitor policy" to see if any of the services is out of policy with local laws and their intentions as appropriate

k. Out of policy notifications are triggered by J and/or automatically from back ground services

l. Out of policy events may have automated remediation (configured by user or default), OR manual intervention by the user.

Curation is the process of keeping content fresh, relevant, and accurate. A curator **108** (e.g., a person operating a machine on a network) may now and again provide content updates, which the system will distribute intelligently across the various content sources in a consistent manner. Thus, for example, a single update to a person's job description may be intelligently normalized and distributed across multiple social media sites and job posting sites. The normalizer **110** receives content updates signal from the curator **108** and in response maps the content to the specific form of the source to which it will be provided (**412**). From there the content is communicated for distribution to (potentially multiple) sources. The system may enable the curator to choose what specific information to view at any time, such as view credit score only, (with the source of the data)

A system consistent with the described embodiment may be utilized to implement a "persomation" app for a PC, Mac or mobile device, either as a standalone app or from within a browser or other app. This app may provide a current and unified view and management portal for a representation of the data representing a person (personal data) from all over the Internet. The data includes and is not limited to: personal financial, demographic, and preference information; things a person likes and dislikes, things a person wishes to purchase, sell, rent, offer for rent, and time frames for such things, etc. Likes and dislikes (whitelist and black list) includes products, services, brand names, companies, organizations, public figures, artist entities such as bands, troupes, artists, entertainers, etc. The person can view and manage the data, either in Tree form (with typical sorting views of, typical taxonomys), or a 2.5D (simulated 3D) interface where the data looks like spheres or recognizable objects, where the size of the object is strength of the relevance score (how personally relevant is the data to the person). Objects are connected to object based on configurable relevancy (e.g. slider controls for people, places, things, events, finance, retail . . . etc.)

DRAWINGS

The system comprises content **108**, content **106**, Collector **112**, curator **116**, Normalizer **104**, Fraud detector **114**, Exploit analyzer **110**, and Certifier **118**. The Collector **112** receives content from content sources **106**, **108** and in response collects the content for normalization and distribution to various management modules. Content may be collected into a buffer, for example a FIFO, or may be processed immediately, or may be buffered and prioritized for processing according to source. Content sources include local devices (PC, LAN server, tablet, mobile phone, etc.), "cloud" locations, and any source that stores data components representing a person or entity explicitly or in an obfuscated fashion, (e.g. GUID).

Content sources may be accessed utilizing a driver model. The system may include specific data source "connectors" built on generic interfaces that call the most optimal methods for gaining access to the source(s), then collecting/distributing data back. For example, local file and/or database methods with local access controls may be utilized for locally access able data stores, e.g. mounted file systems. In the case of accessing personal data on a social network, ecommerce site or financial data warehouse, specific data connectors are built, and loaded, utilizing methods such as REST, JSON, SOAP, Protocol Buffers, and others as appropriate to verify credentials, collect content and distribute updates, etc. Some sources may require subscriptions or some other secure method to access, For example a person may pay also store personal information such as financial accounts and passwords in an encrypted removable device, such as to make it virtually un-hackable.

In some embodiments, a Collector **112** receives a unique human-associated digital identifier from content **106** and transforms it into a digital content. Data triangulator **122** receives a non-unique digital pattern from content **108** and correlates identified data to existing data and transforms a non-unique digital pattern into a unique human-associated digital identifier. Collector **112** transforms personal data signals into a digital content which it transmits to Normalizer **104**.

Normalizer **104** receives a digital content and transforms it into a normalized contents by identifying content attributes and standardizing data formatting. Normalization involves mapping data fields having possibly different names or values across different content sources to consistent names or values for analysis and processing by the various management modules. The normalized content is then distributed to the various management modules.

Certifier **118** receives a normalized content from the Normalizer **104** and in response transforms it into a content certification and transmits the content certification to the Content controller **120**. The Content controller **120** receives this alert from the Certifier **118** and in response maps the alert to the specific form of the source to which it will be provided, such a User interface **124** which presents a person (principle or delegate) with an option to certify the veracity of the content. If the content is validly associated with the person, they may provide an alert indicating its validity. The alert may be passed back to the appropriate content sources which provide an indication that the content is certified by its "owner", that is, the person it is for/about. If the content is not validly associated with the person, a different alert (indicating the content is uncertified) may be generated and passed through to the content sources, which post a corresponding "uncertified" indication for the content.

Exploit analyzer **110** receives a normalized content and in response analyzes the content for exploitation of personal data and transforms it into a first control signal and transmits it to a Content controller **120**. Exploitation may involve use of personal data for monetary gain, without authorization to do so. Detected exploitation may be reported to the person whose data is being exploited using a machine-human interface. The person may choose to act to end the exploitation by contacting the content source or the person exploiting the content.

Fraud detector **114** receives a normalized content and in response analyzes the content for fraudulent actions and transforms it into a first control signal and transmits it to a Content controller **120**. Fraudulent activity can include false attribution of the content to a person or entity; false or misleading statements in conjunction with the content; or other use of the content in a fraudulent or misleading manner. If fraud is detected, an alert may be generated for distribution back to the content source. The Gateway **102** receives the alert from the fraud detector **112** and in response maps the alert to the specific form of the source to which it will be provided. The distributor **126** receives the alert from the Content controller **120** and in response distributes the alert to the appropriate content sources. The content sources **106**, **108** receive the content update signal from distributor **126** which may include an alert and in response act on the alert, for example by removing or flagging the corresponding content.

Curator **116** receives a user content from a User interface **124** and transforms the digital content into a third control signal and transmits it to Content controller **120**.

Content controller **120** receives a third control signal, a second control signal, a first control signal and a content certification and sends a combined control signal to Gateway **102** operate Gateway **102** and transmit an enable signal to Collector **112**.

Content controller **120** receives a third control signal, a second control signal, a first control signal and a content certification and sends an alert to User interface **124** to notify user of the need for additional action or approval.

Distributor **126** receives an approved content and alert and sends updates and alerts signal to content **106** and content **108**.

In block **202**, routine **200** transforms non-unique data signals into a unique human-associated digital identifier.

In block **204**, routine **200** transforms the unique human-associated digital identifier into a digital content.

In block **206**, routine **200** normalizes the digital content into a plurality of normalized content.

In block **208**, routine **200** transforms at least one of the normalized content into an enable signal.

In block **210**, routine **200** transforms at least one of the normalized content into a second control signal.

In block **212**, routine **200** transforms at least one of the normalized content into an alert.

In block **214**, routine **200** transforms a user content into a third control signal.

In block **216**, routine **200** transforms at least one of the first control signal.

In block **218**, routine **200** transforms at least one of the first control signal.

In block **220**, routine **200** combines a normalized content with a combined control signal to control an approved content gateway and generate an enable signal.

In block **222**, routine **200** transforms an enable signal and alert into an updates and alerts signal.

In done block **224**, routine **200** ends.

In block **306**, the Data triangulator **122** receives a non-unique digital pattern from the content **108** and transforms it into a unique human-associated digital identifier.

In block **304**, collector **112** receives a unique human-associated digital identifier from data triangulator **122** and in response collects the content from multiple sources of content

In block **302**, collector **112** receives a unique human-associated digital identifier from the content source **106** and in response collects the content from multiple sources of content.

In block **308**, the normalizer **104** receives a digital content from the Collector **112** and in response performs normalization on the content by identifying content attributes and mapping data fields having possibly different names or values across different content sources to consistent names or values for analysis and processing by the various management modules.

In block **310**, the fraud detector **114** receives a digital content from the normalizer **104** and in response analyzes the content for fraudulent actions.

In block **312**, the exploit analyzer **110** receives a digital content from the normalizer **104** and in response analyzes the content for exploitation of personal data.

In block **314**, the certifier **118** receives a digital content from the normalizer **104** and in response presents a person (principle or delegate) with an option to certify the veracity of the content.

In block **402**, the fraud detector **114** receives a content signal from the normalizer **104** and in response analyzes the content for fraudulent actions.

In block **404**, the exploit analyzer **110** receives digital content from the normalizer **104** and in response analyzes the content for the exploitation of personal data.

In block **406**, the certifier **118** receives a digital content from the normalizer **110** and in response presents a person (principle or delegate) with an option to certify the veracity of the content.

In block **408**, the content controller **120** receives an alert from the Fraud detector **114** and in response maps the alert to the source to which it will be provided and transmits a combined control signal to Gateway **102**

In block **414**, the content Content controller **120** receives a first control signal from the exploit analyzer **110** and in response in response maps the alert to the source to which it will be provided and transmits a combined control signal to Gateway **102**.

In block **412**, the Content controller **120** receives a content certification from the certifier **118** and in response maps the alert to the source to which it will be provided and transmits a combined control signal to Gateway **102**.

In block **410**, The content controller **120** receives a third control signal from curator **116** and in response maps the content to the source to which it will be provided.

In block **416**, the distributor **126** receives an updates and alerts signal from the content controller **120** and in response distributes the updates and alerts to the appropriate content source.

In block **502**, the distributor **126** receives an updates and alerts signal from the content controller **120** and in response distributes the updates and alerts to the appropriate content source.

In block **504** the content source **106** receives an updates and alerts signal from the distributor and in response updates the corresponding content or acts on the alert.

In block **506** the content source **108** receives an updates and alerts signal from the distributor and in response updates the corresponding content or acts on the alert.

FIG. **6** illustrates an embodiment of a digital apparatus **600** to implement components and process steps of the system described herein.

Input devices **604** comprise transducers that convert physical phenomenon into machine internal signals, typically electrical, optical or magnetic signals. Signals may also be wireless in the form of electromagnetic radiation in the radio frequency (RF) range but also potentially in the infrared or optical range. Examples of input devices **604** are keyboards which respond to touch or physical pressure from an object or proximity of an object to a surface, mice which respond to motion through space or across a plane, microphones which convert vibrations in the medium (typically air) into device signals, scanners which convert optical patterns on two or three dimensional objects into device signals. The signals from the input devices **604** are provided via various machine signal conductors (e.g., busses or network interfaces) and circuits to memory **606**.

The memory **606** is typically what is known as a first or second level memory device, providing for storage (via configuration of matter or states of matter) of signals received from the input devices **604**, instructions and information for controlling operation of the CPU **602**, and signals from storage devices **610**.

Information stored in the memory **606** is typically directly accessible to the CPU **602** of the device. Signals input to the device cause the reconfiguration of the internal material/energy state of the memory **606**, creating in essence a new machine configuration, influencing the behavior of the digital apparatus **600** by affecting the behavior of the CPU **602** with control signals (instructions) and data provided in conjunction with the control signals.

Second or third level storage devices **610** may provide a slower but higher capacity machine memory capability. Examples of storage devices **610** are hard disks, optical disks, large capacity flash memories or other non-volatile memory technologies, and magnetic memories.

The CPU **602** may cause the configuration of the memory **606** to be altered by signals in storage devices **610**. In other words, the CPU **602** may cause data and instructions to be read from storage devices **610** in the memory **606** from which may then influence the operations of CPU **602** as instructions and data signals, and from which it may also be provided to the output devices **608**. The CPU **602** may alter the content of the memory **606** by signaling to a machine interface of memory **606** to alter the internal configuration, and then converted signals to the storage devices **610** to alter its material internal configuration. In other words, data and instructions may be backed up from memory **606**, which is often volatile, to storage devices **610**, which are often non-volatile.

Output devices **608** are transducers which convert signals received from the memory **606** into physical phenomenon such as vibrations in the air, or patterns of light on a machine display, or vibrations (i.e., haptic devices) or patterns of ink or other materials (i.e., printers and 3-D printers).

The network interface **612** receives signals from the memory **606** and converts them into electrical, optical, or wireless signals to other machines, typically via a machine network. The network interface **612** also receives signals from the machine network and converts them into electrical, optical, or wireless signals to the memory **606**.

FIG. **7** illustrates several components of an exemplary server **700** in accordance with one embodiment. In various embodiments, server **700** may include a desktop PC, server, workstation, mobile phone, laptop, tablet, set-top box, appliance, or other computing device that is capable of performing operations such as those described herein. In some embodiments, server **700** may include many more components than those shown in FIG. **7**. However, it is not necessary that all of these generally conventional components be shown in order to disclose an illustrative embodiment. Collectively, the various tangible components or a subset of the tangible components may be referred to herein as "logic" configured or adapted in a particular way, for example as logic configured or adapted with particular software or firmware.

In various embodiments, server **700** may comprise one or more physical and/or logical devices that collectively provide the functionalities described herein. In some embodiments, server **700** may comprise one or more replicated and/or distributed physical or logical devices.

In some embodiments, server **700** may comprise one or more computing resources provisioned from a "cloud computing" provider, for example, Amazon Elastic Compute Cloud ("Amazon EC2"), provided by Amazon.com, Inc. of Seattle, Wash.; Sun Cloud Compute Utility, provided by Sun Microsystems, Inc. of Santa Clara, Calif.; Windows Azure, provided by Microsoft Corporation of Redmond, Wash., and the like.

Server **700** includes a bus **702** interconnecting several components including a network interface **708**, a display **706**, a central processing unit **710**, and a memory **704**.

Memory **704** generally comprises a random access memory ("RAM") and permanent non-transitory mass storage device, such as a hard disk drive or solid-state drive. Memory **704** stores an operating system **712**.

These and other software components may be loaded into memory **704** of server **700** using a drive mechanism (not shown) associated with a non-transitory computer-readable medium **716**, such as a floppy disc, tape, DVD/CD-ROM drive, memory card, or the like.

Memory **704** also includes database **714**. In some embodiments, server **200** (deleted) may communicate with database **714** via network interface **708**, a storage area network ("SAN"), a high-speed serial bus, and/or via the other suitable communication technology.

In some embodiments, database **714** may comprise one or more storage resources provisioned from a "cloud storage" provider, for example, Amazon Simple Storage Service ("Amazon S3"), provided by Amazon.com, Inc. of Seattle, Wash., Google Cloud Storage, provided by Google, Inc. of Mountain View, Calif., and the like.

References to "one embodiment" or "an embodiment" do not necessarily refer to the same embodiment, although they may. Unless the context clearly requires otherwise, throughout the description and the claims, the words "comprise," "comprising," and the like are to be construed in an inclusive sense as opposed to an exclusive or exhaustive sense; that is to say, in the sense of "including, but not limited to." Words using the singular or plural number also include the plural or singular number respectively, unless expressly limited to a single one or multiple ones. Additionally, the words "herein," "above," "below" and words of similar import, when used in this application, refer to this application as a whole and not to any particular portions of this application. When the claims use the word "or" in reference to a list of two or more items, that word covers all of the following interpretations of the word: any of the items in the list, all of the items in the list and any combination of the items in the list, unless expressly limited to one or the other.

"Logic" refers to machine memory circuits, non transitory machine readable media, and/or circuitry which by way of its material and/or material-energy configuration comprises control and/or procedural signals, and/or settings and values (such as resistance, impedance, capacitance, inductance, current/voltage ratings, etc.), that may be applied to influence the operation of a device. Magnetic media, electronic circuits, electrical and optical memory (both volatile and nonvolatile), and firmware are examples of logic. Logic specifically excludes pure signals or software per se (however does not exclude machine memories comprising software and thereby forming configurations of matter).

Those skilled in the art will appreciate that logic may be distributed throughout one or more devices, and/or may be comprised of combinations memory, media, processing circuits and controllers, other circuits, and so on. Therefore, in the interest of clarity and correctness logic may not always be distinctly illustrated in drawings of devices and systems, although it is inherently present therein.

The techniques and procedures described herein may be implemented via logic distributed in one or more computing devices. The particular distribution and choice of logic will vary according to implementation.

Those having skill in the art will appreciate that there are various logic implementations by which processes and/or systems described herein can be effected (e.g., hardware, software, and/or firmware), and that the preferred vehicle will vary with the context in which the processes are deployed. "Software" refers to logic that may be readily readapted to different purposes (e.g. read/write volatile or nonvolatile memory or media). "Firmware" refers to logic embodied as read-only memories and/or media. Hardware refers to logic embodied as analog and/or digital circuits. If an implementer determines that speed and accuracy are paramount, the implementer may opt for a hardware and/or firmware vehicle; alternatively, if flexibility is paramount, the implementer may opt for a solely software implementation; or, yet again alternatively, the implementer may opt for some combination of hardware, software, and/or firmware. Hence, there are several possible vehicles by which the processes described herein may be effected, none of which is inherently superior to the other in that any vehicle to be utilized is a choice dependent upon the context in which the vehicle will be deployed and the specific concerns (e.g., speed, flexibility, or predictability) of the implementer, any of which may vary. Those skilled in the art will recognize that optical aspects of implementations may involve optically-oriented hardware, software, and or firmware.

The foregoing detailed description has set forth various embodiments of the devices and/or processes via the use of block diagrams, flowcharts, and/or examples. Insofar as such block diagrams, flowcharts, and/or examples contain one or more functions and/or operations, it will be understood as notorious by those within the art that each function and/or operation within such block diagrams, flowcharts, or examples can be implemented, individually and/or collectively, by a wide range of hardware, software, firmware, or virtually any combination thereof. Several portions of the subject matter described herein may be implemented via Application Specific Integrated Circuits (ASICs), Field Programmable Gate Arrays (FPGAs), digital signal processors (DSPs), or other integrated formats. However, those skilled in the art will recognize that some aspects of the embodiments disclosed herein, in whole or in part, can be equivalently implemented in standard integrated circuits, as one or more computer programs running on one or more computers (e.g., as one or more programs running on one or more computer systems), as one or more programs running on one or more processors (e.g., as one or more programs running on one or more microprocessors), as firmware, or as virtually any combination thereof, and that designing the circuitry and/or writing the code for the software and/or firmware would be well within the skill of one of skill in the art in light of this disclosure. In addition, those skilled in the art will appreciate that the mechanisms of the subject matter described herein are capable of being distributed as a program product in a variety of forms, and that an illustrative embodiment of the subject matter described herein applies equally regardless of the particular type of signal bearing media used to actually carry out the distribution. Examples of a signal bearing media include, but are not limited to, the following: recordable type media such as floppy disks, hard disk drives, CD ROMs, digital tape, flash drives, SD cards, solid state fixed or removable storage, and computer memory.

In a general sense, those skilled in the art will recognize that the various aspects described herein which can be implemented, individually and/or collectively, by a wide range of hardware, software, firmware, or any combination thereof can be viewed as being composed of various types of "circuitry." Consequently, as used herein "circuitry" includes, but is not limited to, electrical circuitry having at least one discrete electrical circuit, electrical circuitry having at least one integrated circuit, electrical circuitry having at least one application specific integrated circuit, circuitry forming a general purpose computing device configured by a computer program (e.g., a general purpose computer configured by a computer program which at least partially carries out processes and/or devices described herein, or a microprocessor configured by a computer program which at least partially carries out processes and/or devices described herein), circuitry forming a memory device (e.g., forms of random access memory), and/or circuitry forming a communications device (e.g., a modem, communications switch, or optical-electrical equipment).

Those skilled in the art will recognize that it is common within the art to describe devices and/or processes in the fashion set forth herein, and thereafter use standard engineering practices to integrate such described devices and/or processes into larger systems. That is, at least a portion of the devices and/or processes described herein can be integrated into a network processing system via a reasonable amount of experimentation.

What is claimed is:

1. A system comprising:

machine-readable instructions that when executed by one or more computer processors, configure the system to:

transform a non-unique digital pattern into a unique human-associated digital identifier;

collect digital content associated with said unique human-associated digital identifier;

transform said digital content into a plurality of normalized content, each normalized content comprising a format of the digital content adapted to a different destination web site on which to post the normalized content;

generate a content certification for the normalized content;

apply a data policy comprising rules for data usage by a user uniquely associated with the digital identifier to the normalized content to generate a first control signal;

analyze said normalized content for fraudulent activity to generate a second control signal;

receive additional content from a machine user interface and curate the additional content into a third control signal;

controller logic activated by said third control signal, said content certification, said first control signal, and said second control signal to generate an alert and a combined control signal;

a digital gateway coupled to the controller logic to transform said combined control signal into an enable signal to a digital content distributor; and

the digital content distributor activated by said enable signal and said alert to modify the normalized content with the additional content and transmit the modified normalized content to the different destination web sites.

2. The system of claim **1** wherein transforming said digital content into the plurality of normalized content comprises receiving the digital content and converting said digital content into the normalized content with standardized constituent personal data attributes.

3. The system of claim **1** wherein analyzing said normalized content for fraudulent activity further comprises comparing the normalized content to a plurality of data attributes for the user.

4. The system of claim **1** further comprising machine-readable instructions that when executed by the one or more computer processors, configure the system to identify personal information for the user from a digital content source.

5. The system of claim **1** wherein transforming the non-unique digital pattern into the unique human-associated digital identifier further comprises:

parsing the non-unique digital pattern into at least one non-unique personal data attributes; attribute;

comparing said at least one non-unique personal data attribute to user personal data attributes; and

compiling the user personal data attributes into the unique human-associated digital identifier.

* * * * *

RoboReview™ Date prepared: July 24th 2018, 11:27:11

File name: US9619661B1.docx

Art Unit Predictions

Statistics for the five most-likely results, in decreasing order

Art Unit	Allowance rate	Pendency (months)	Avg. no. of Office actions	% granted with appeal
2459	43%	51	3.3	14%
2423	72%	35	2.5	10%
3621	19%	66	4.0	23%
3684	47%	50	3.3	16%
2498	84%	32	2.0	5%

§101

Eligibility based on similarity to claims rejected under 101 for abstraction

Low ELIGIBILITY High

Words related to low eligibility:

normalized *transform* *policy* *into*
for *certification*

§102 §103

Novelty based on most-similar pieces of art

Related patent documents

9619661 7444306 9002744 20130097009
20160026341 20050091164 7895123 6389538
6993508 20010056460

§112

Clarity issues based on language defects in the application

0	0	0	0
Antecedent basis issues	Figure reference issues	Unsupported claim terms	Claim order and format issues

Antecedent-basis issues

RoboReview™ found no antecedent-basis issues

Claim support issues

RoboReview™ found no claim support issues

Claim number and format

RoboReview™ found no claim number and format issues

issues

Parts list

RoboReview™ found no issues with numbered parts

TURBOPATENT®

TurboPatent Sample Patent

AN AUTOMATIC ELECTRICAL SIGNAL DECODING DISCOVERY AND ELECTRICAL SIGNAL CODING BETWEEN A DEVICE AND ITS REMOTE CONTROLLER

US 9,774,480

September 26, 2017

US009774480B1

(12) United States Patent
Rougier

(10) Patent No.: US 9,774,480 B1
(45) Date of Patent: Sep. 26, 2017

(54) **AUTOMATIC ELECTRICAL SIGNAL DECODING DISCOVERY AND ELECTRICAL SIGNAL CODING BETWEEN A DEVICE AND ITS REMOTE CONTROLLER**

(71) Applicant: **Kirio Inc.**, Seattle, WA (US)

(72) Inventor: **Franck D Rougier**, Seattle, WA (US)

(73) Assignee: **Kirio Inc.**, Seattle, WA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **15/265,099**

(22) Filed: **Sep. 14, 2016**

Related U.S. Application Data

(60) Provisional application No. 62/218,379, filed on Sep. 14, 2015.

(51) **Int. Cl.**
H04L 27/00 (2006.01)

(52) **U.S. Cl.**
CPC ***H04L 27/0012*** (2013.01); ***H04L 27/0008*** (2013.01); *H04L 27/0002* (2013.01)

(58) **Field of Classification Search**
CPC H04L 27/0012; H04L 27/0008; H04L 27/0002
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

2003/0053794 A1* 3/2003 Krouglov G11B 20/02 386/275
2010/0219808 A1* 9/2010 Steckley G01R 19/2513 324/76.77

OTHER PUBLICATIONS

Ka Mun et al("Automatic recognition and demodulation of digitally modulated communications signals using wavelet-domain signatures" Graduate School—New Brunswick, New Jersey, Jan. 2010, pp. 1-200.*

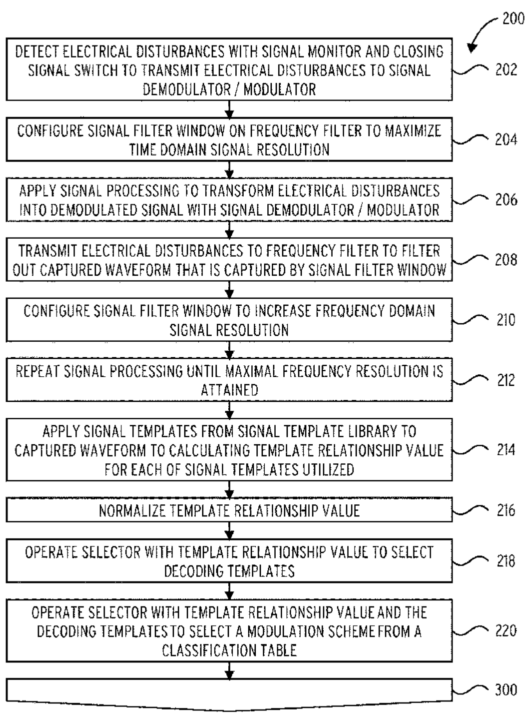
* cited by examiner

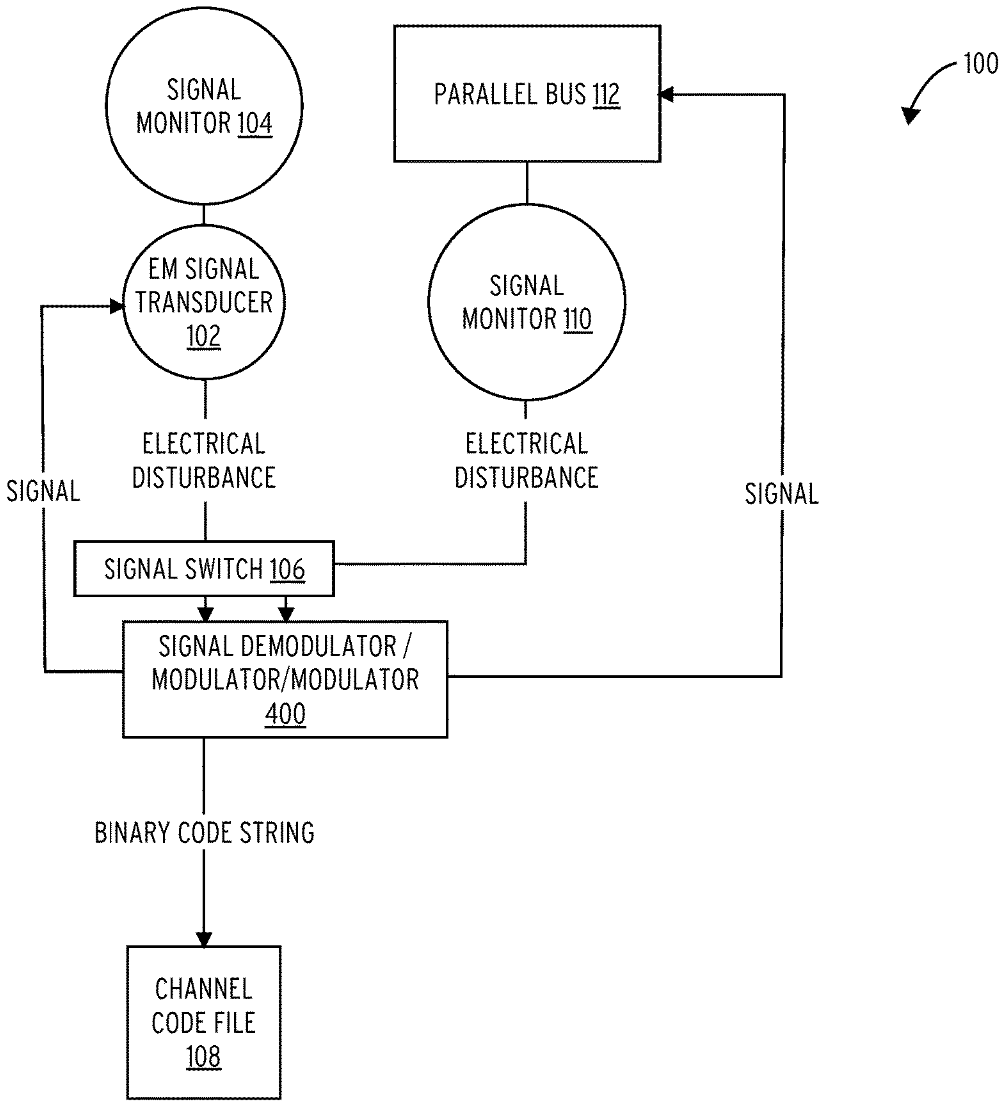
Primary Examiner — Helene Tayong
(74) *Attorney, Agent, or Firm* — FSP LLC

(57) **ABSTRACT**

A process for automatic electrical signal discovery and decoding includes detecting electrical disturbances in a serial or parallel bus, or from a transducer with a signal monitor and demodulating the signals by configuring a signal filter window on a frequency filter to maximize time domain signal resolution. Filtering out each frequency captured by the filter and adapting the signal filter window to increase frequency resolution until maximal frequency resolution is attained, demodulating the signal based on the detected encoding and transforming the demodulated signal into a binary code string for transcription to a channel code file.

16 Claims, 9 Drawing Sheets





100
SIGNAL MONITOR 104
PARALLEL BUS 112
EM SIGNAL TRANSDUCER 102
SIGNAL MONITOR 110
SIGNAL
ELECTRICAL DISTURBANCE
ELECTRICAL DISTURBANCE
SIGNAL
SIGNAL SWITCH 106
SIGNAL DEMODULATOR / /MODULATOR/MODULATOR 400
BINARY CODE STRING
CHANNEL CODE FILE 108

FIG. 1

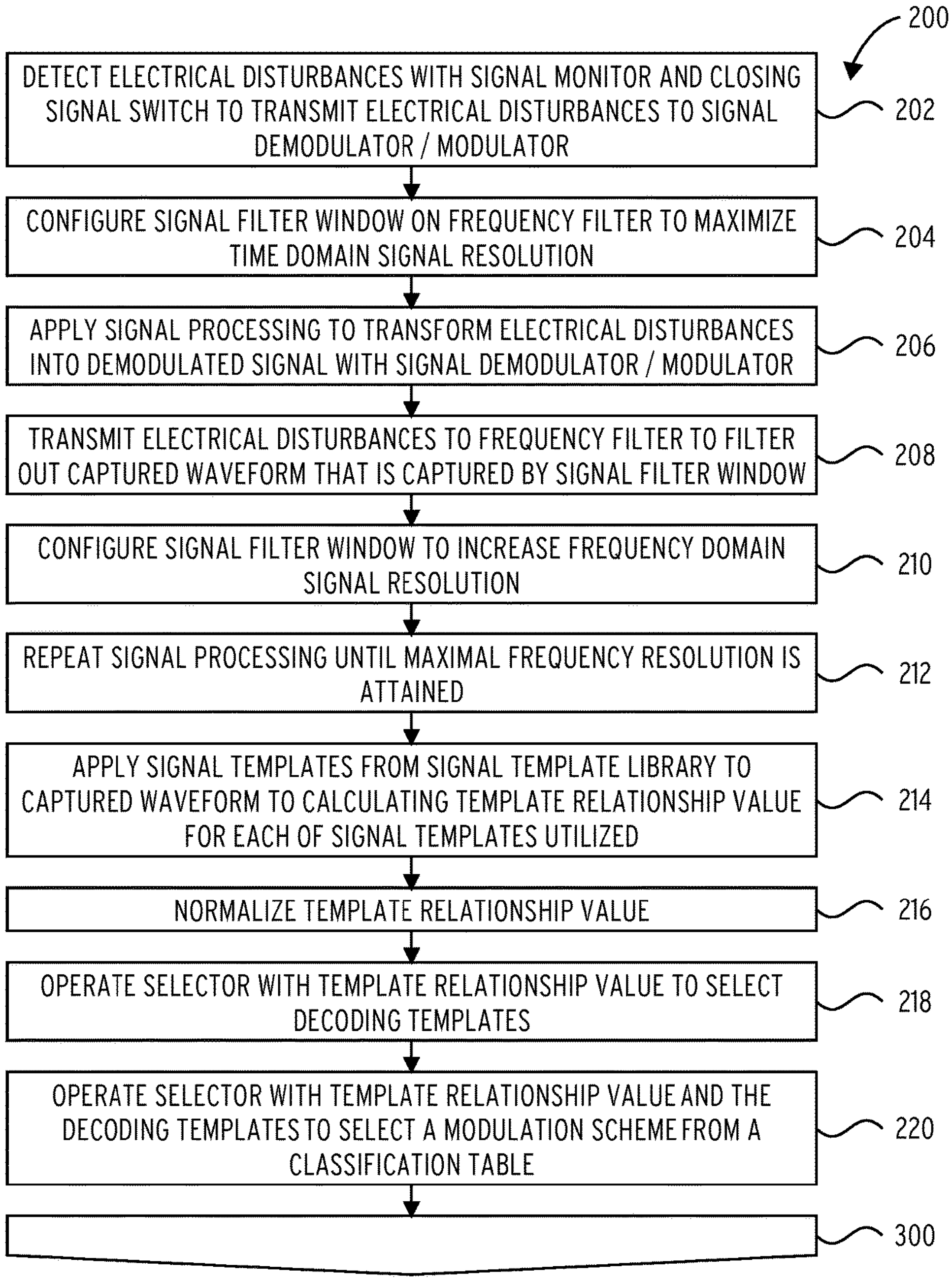

200
DETECT ELECTRICAL DISTURBANCES WITH SIGNAL MONITOR AND CLOSING SIGNAL SWITCH TO TRANSMIT ELECTRICAL DISTURBANCES TO SIGNAL DEMODULATOR / MODULATOR
202
CONFIGURE SIGNAL FILTER WINDOW ON FREQUENCY FILTER TO MAXIMIZE TIME DOMAIN SIGNAL RESOLUTION
204
APPLY SIGNAL PROCESSING TO TRANSFORM ELECTRICAL DISTURBANCES INTO DEMODULATED SIGNAL WITH SIGNAL DEMODULATOR / MODULATOR
206
TRANSMIT ELECTRICAL DISTURBANCES TO FREQUENCY FILTER TO FILTER OUT CAPTURED WAVEFORM THAT IS CAPTURED BY SIGNAL FILTER WINDOW
208
CONFIGURE SIGNAL FILTER WINDOW TO INCREASE FREQUENCY DOMAIN SIGNAL RESOLUTION
210
REPEAT SIGNAL PROCESSING UNTIL MAXIMAL FREQUENCY RESOLUTION IS ATTAINED
212
APPLY SIGNAL TEMPLATES FROM SIGNAL TEMPLATE LIBRARY TO CAPTURED WAVEFORM TO CALCULATING TEMPLATE RELATIONSHIP VALUE FOR EACH OF SIGNAL TEMPLATES UTILIZED
214
NORMALIZE TEMPLATE RELATIONSHIP VALUE
216
OPERATE SELECTOR WITH TEMPLATE RELATIONSHIP VALUE TO SELECT DECODING TEMPLATES
218
OPERATE SELECTOR WITH TEMPLATE RELATIONSHIP VALUE AND THE DECODING TEMPLATES TO SELECT A MODULATION SCHEME FROM A CLASSIFICATION TABLE
220
300

FIG. 2

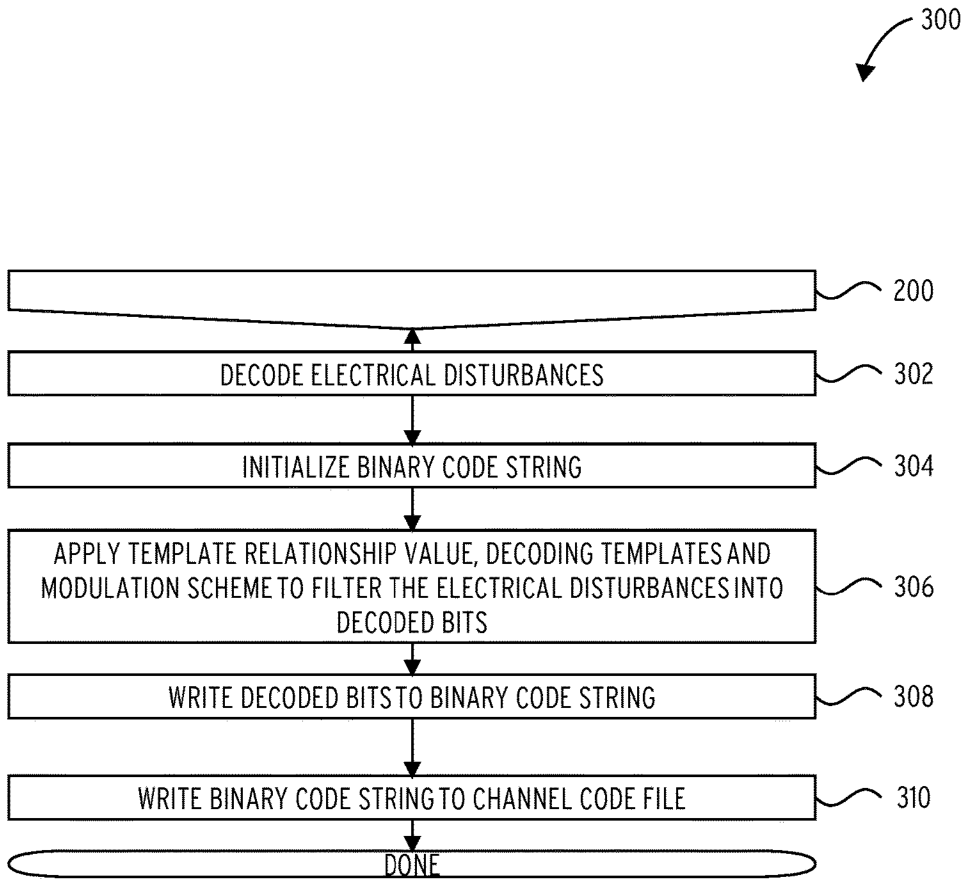

300
200
DECODE ELECTRICAL DISTURBANCES
302
INITIALIZE BINARY CODE STRING
304
APPLY TEMPLATE RELATIONSHIP VALUE, DECODING TEMPLATES AND MODULATION SCHEME TO FILTER THE ELECTRICAL DISTURBANCES INTO DECODED BITS
306
WRITE DECODED BITS TO BINARY CODE STRING
308
WRITE BINARY CODE STRING TO CHANNEL CODE FILE
310
DONE

FIG. 3

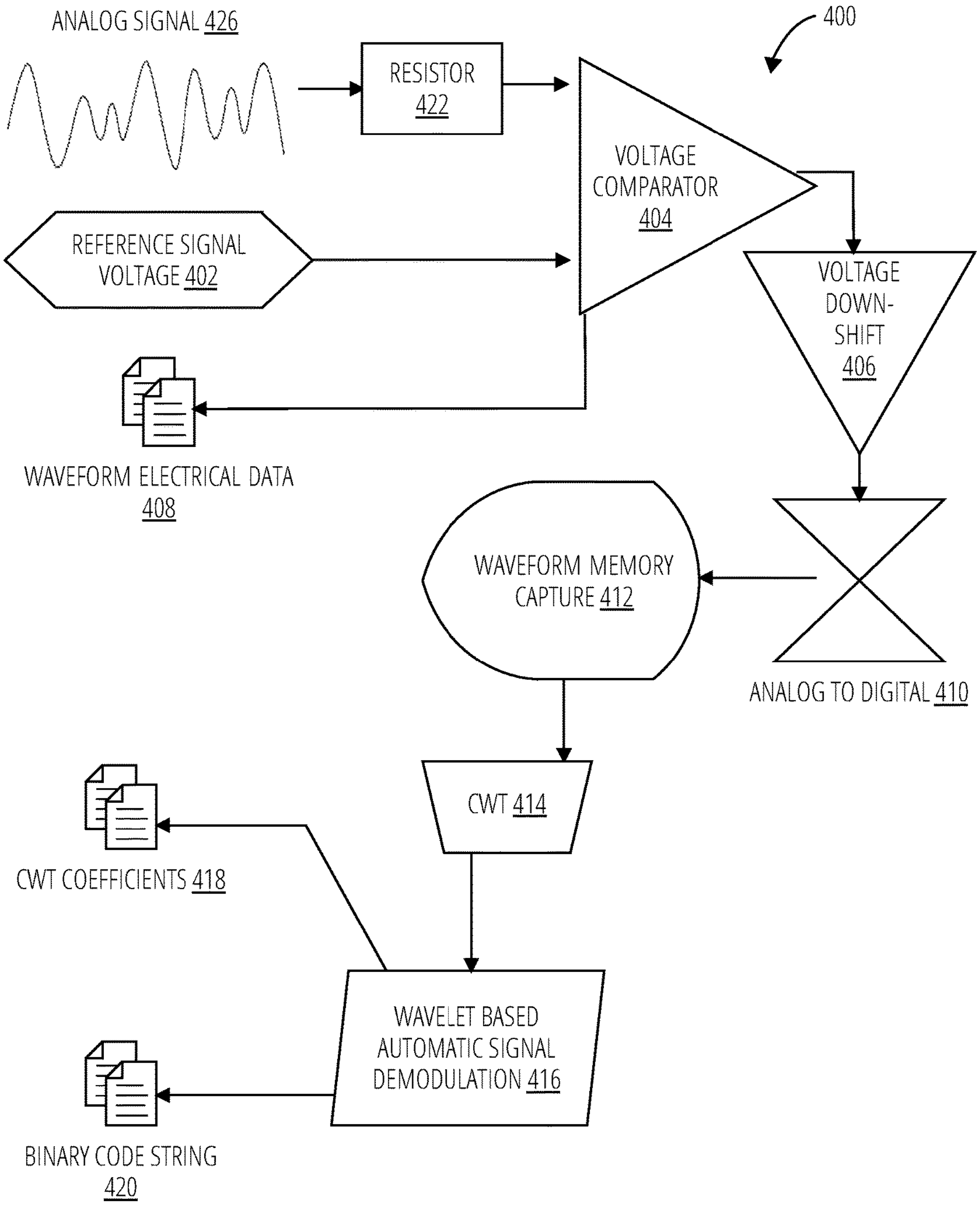

ANALOG SIGNAL 426
RESISTOR 422
400
VOLTAGE COMPARATOR 404
REFERENCE SIGNAL VOLTAGE 402
VOLTAGE DOWN-SHIFT 406
WAVEFORM ELECTRICAL DATA 408
WAVEFORM MEMORY CAPTURE 412
ANALOG TO DIGITAL 410
CWT 414
CWT COEFFICIENTS 418
WAVELET BASED AUTOMATIC SIGNAL DEMODULATION 416
BINARY CODE STRING 420

FIG. 4

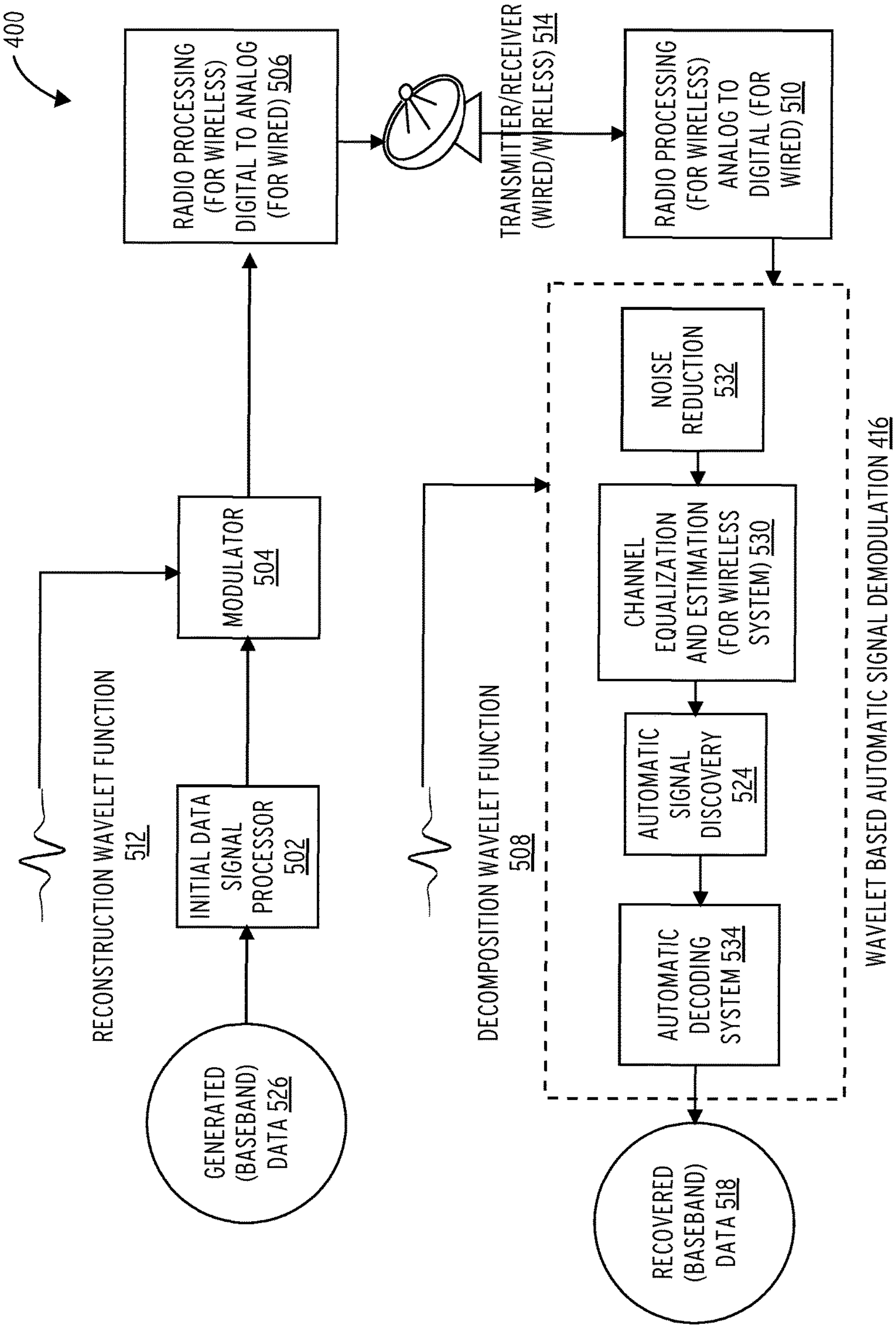

400
GENERATED (BASEBAND) DATA 526
INITIAL DATA SIGNAL PROCESSOR 502
RECONSTRUCTION WAVELET FUNCTION 512
MODULATOR 504
RADIO PROCESSING (FOR WIRELESS) DIGITAL TO ANALOG (FOR WIRED) 506
TRANSMITTER/RECEIVER (WIRED/WIRELESS) 514
RADIO PROCESSING (FOR WIRELESS) ANALOG TO DIGITAL (FOR WIRED) 510
NOISE REDUCTION 532
CHANNEL EQUALIZATION AND ESTIMATION (FOR WIRELESS SYSTEM) 530
AUTOMATIC SIGNAL DISCOVERY 524
AUTOMATIC DECODING SYSTEM 534
DECOMPOSITION WAVELET FUNCTION 508
WAVELET BASED AUTOMATIC SIGNAL DEMODULATION 416
RECOVERED (BASEBAND) DATA 518

FIG. 5

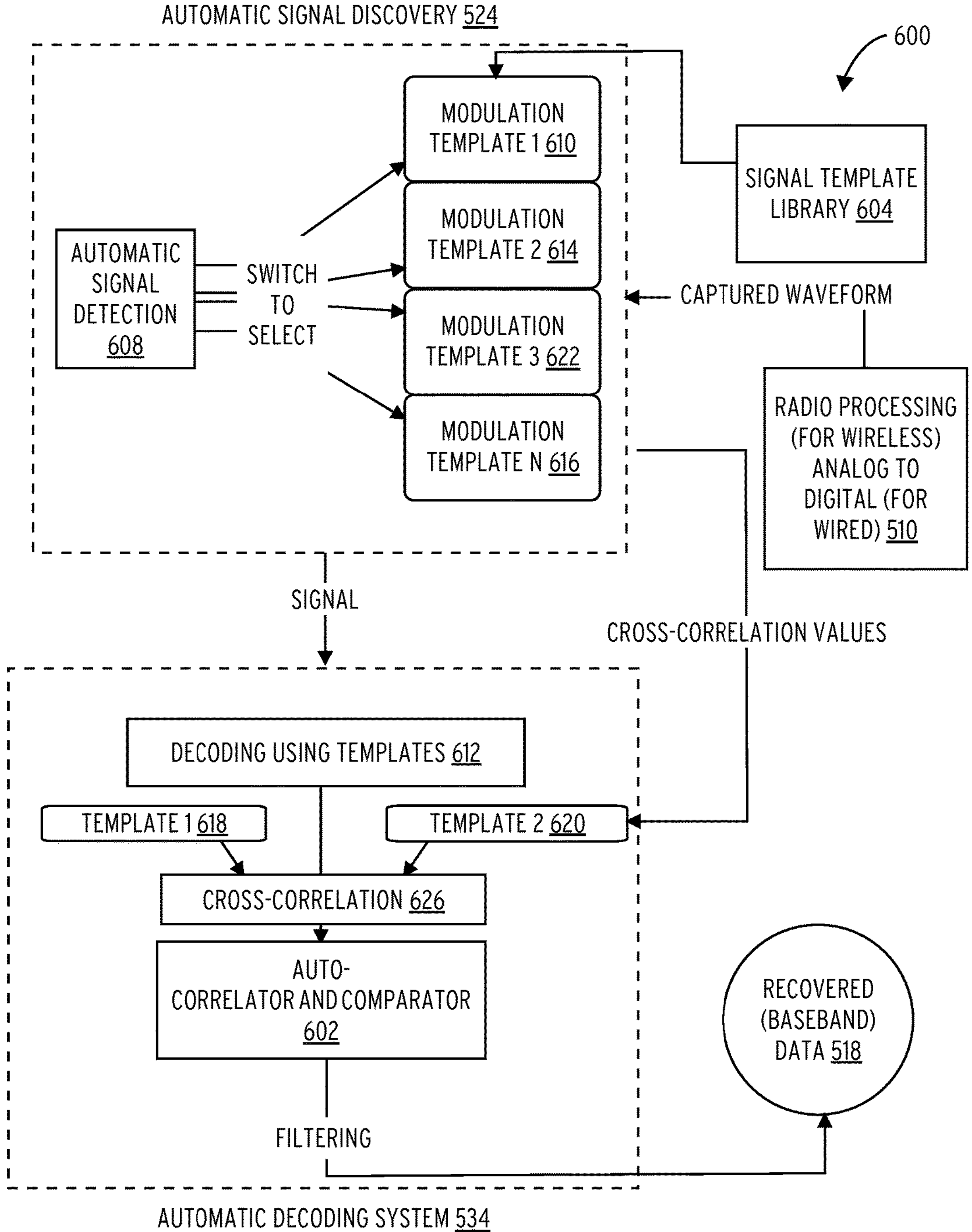

AUTOMATIC SIGNAL DISCOVERY 524
600
MODULATION TEMPLATE 1 610
MODULATION TEMPLATE 2 614
MODULATION TEMPLATE 3 622
MODULATION TEMPLATE N 616
SIGNAL TEMPLATE LIBRARY 604
AUTOMATIC SIGNAL DETECTION 608
SWITCH TO SELECT
CAPTURED WAVEFORM
RADIO PROCESSING (FOR WIRELESS) ANALOG TO DIGITAL (FOR WIRED) 510
SIGNAL
CROSS-CORRELATION VALUES
DECODING USING TEMPLATES 612
TEMPLATE 1 618
TEMPLATE 2 620
CROSS-CORRELATION 626
AUTO-CORRELATOR AND COMPARATOR 602
RECOVERED (BASEBAND) DATA 518
FILTERING
AUTOMATIC DECODING SYSTEM 534

FIG. 6

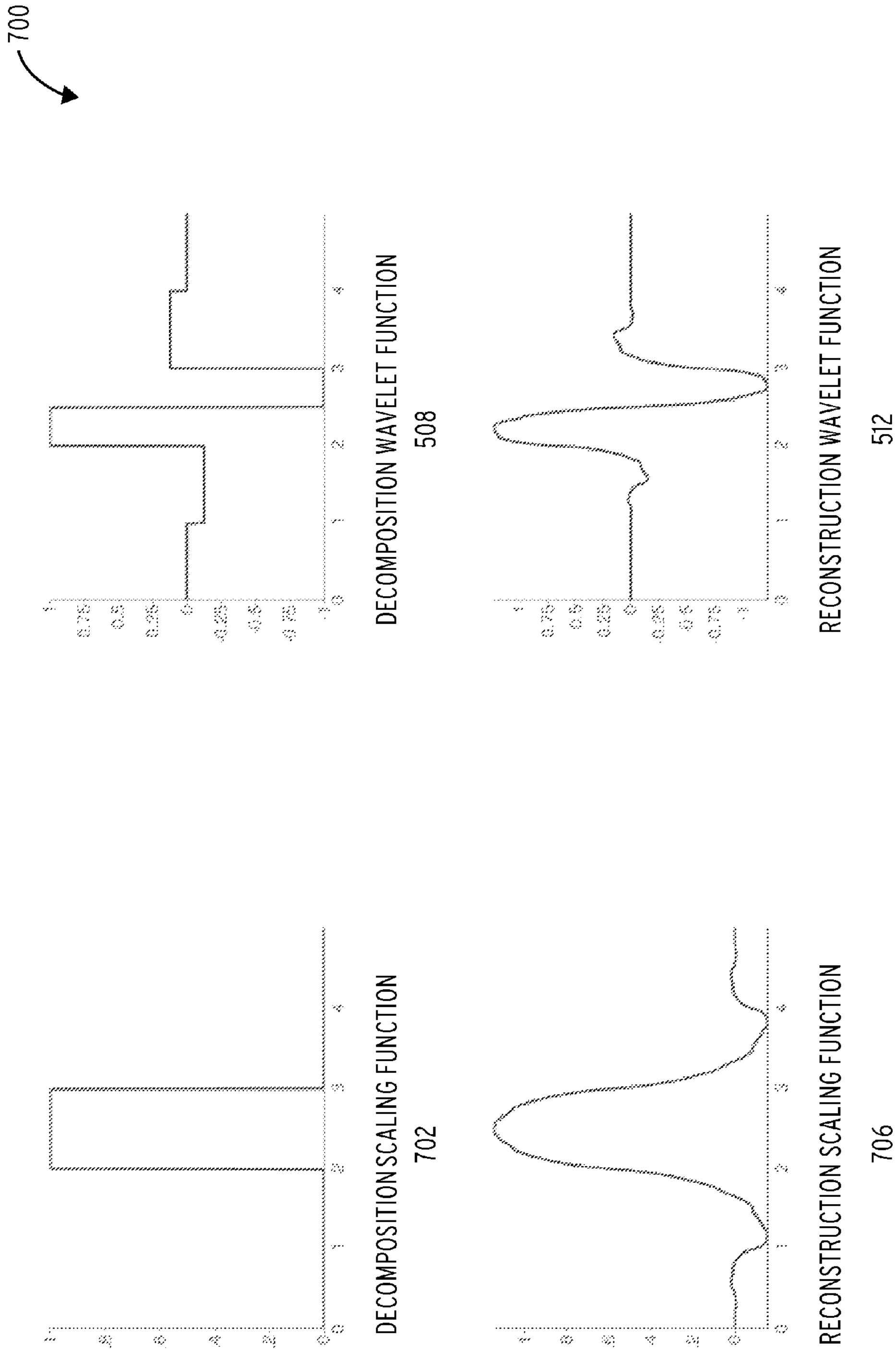

700
DECOMPOSITION SCALING FUNCTION
702
RECONSTRUCTION SCALING FUNCTION
706
DECOMPOSITION WAVELET FUNCTION
508
RECONSTRUCTION WAVELET FUNCTION
512

FIG. 7

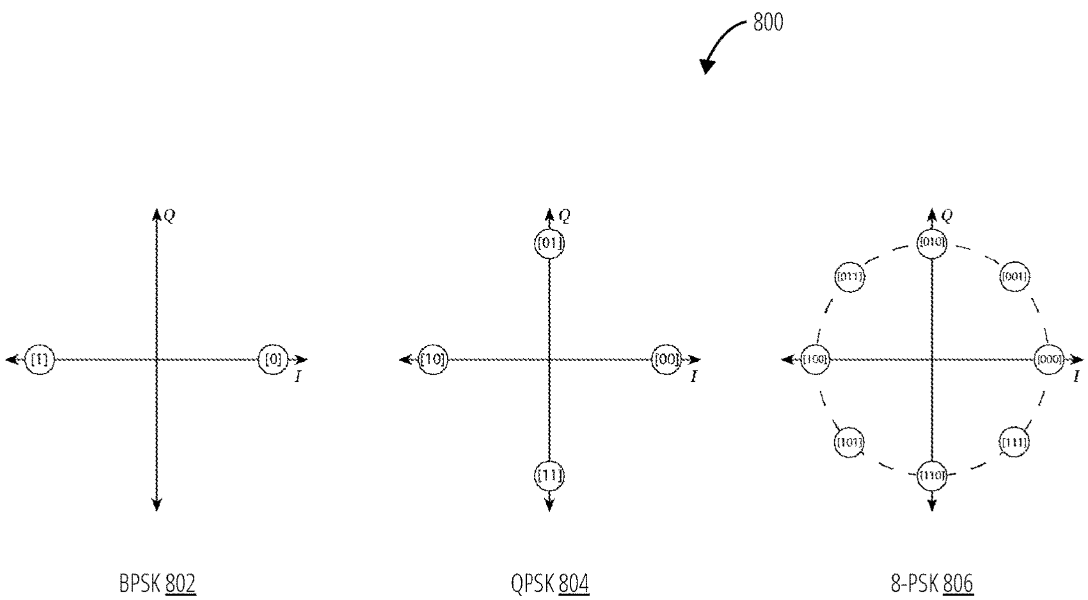
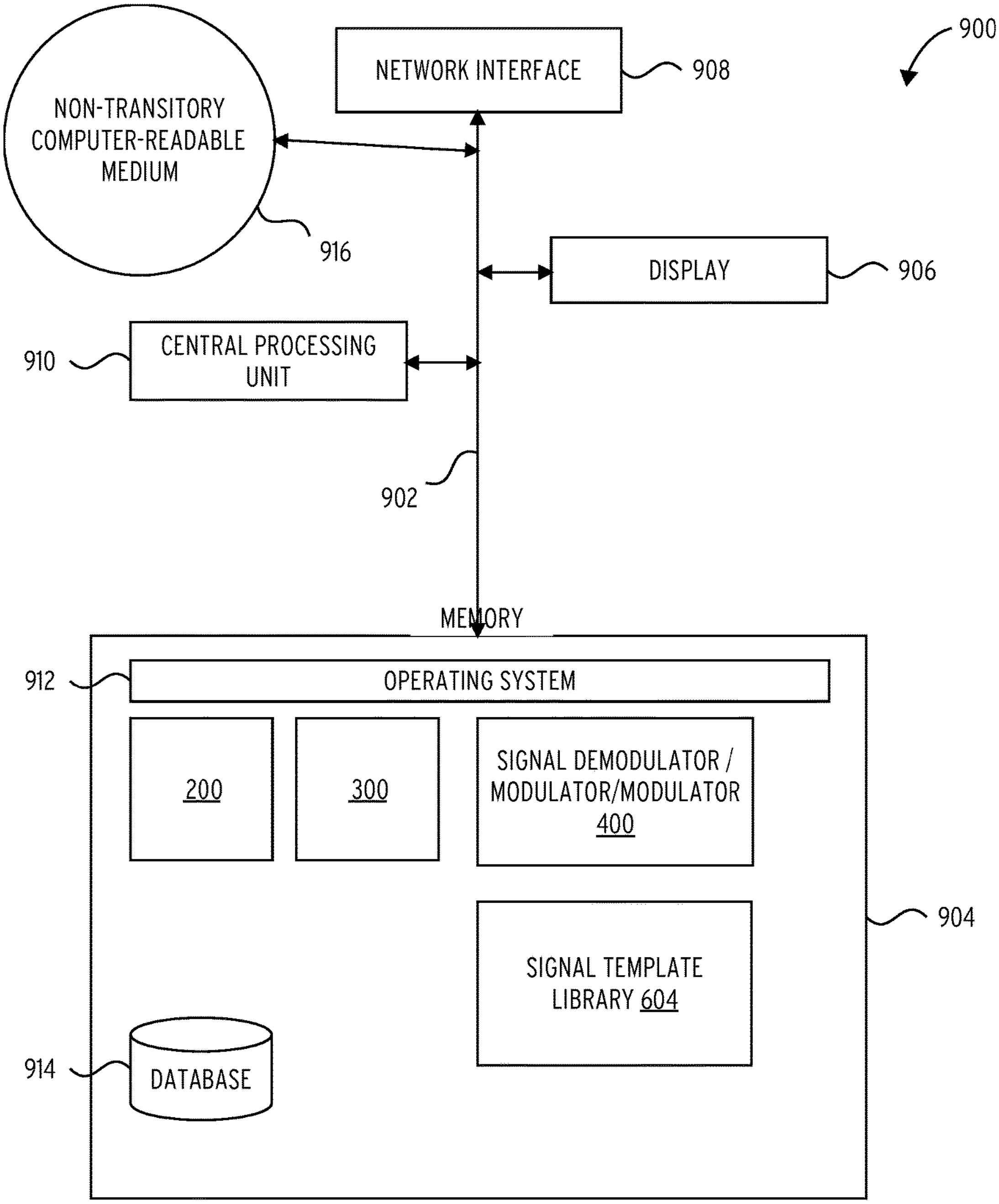

800
Q
[1]
[0]
I
BPSK 802
Q
[01]
[10]
[00]
I
[11]
QPSK 804
Q
[010]
[011]
[001]
[100]
[000]
I
[101]
[111]
[110]
8-PSK 806

FIG. 8



900
NETWORK INTERFACE
908
NON-TRANSITORY COMPUTER-READABLE MEDIUM
916
DISPLAY
906
910
CENTRAL PROCESSING UNIT
902
MEMORY
912
OPERATING SYSTEM
200
300
SIGNAL DEMODULATOR / MODULATOR/MODULATOR 400
904
SIGNAL TEMPLATE LIBRARY 604
914
DATABASE

FIG. 9

AUTOMATIC ELECTRICAL SIGNAL DECODING DISCOVERY AND ELECTRICAL SIGNAL CODING BETWEEN A DEVICE AND ITS REMOTE CONTROLLER

BACKGROUND

Typical modulation/demodulation schemes start with foreknowledge of the intended use, the physical interconnect, and other system requirements. Because every manufacturer uses different approaches in their design, the market has multiple systems to communicate information from a device providing a useful function to its remote controller. Typical universal remote controllers simply record a visible spectra of information sent from a given controller then replay it. These are popular with optical systems such as infrared remote controls. With the advent of modern equipment, information is often exchanged prior to an operation between the device and the remote controller. Therefore, blindly recording the information is often not sufficient. In addition, because the information sent is parameterized (e.g. temperature settings), it is not feasible to emulate all variations of the parameters to arrive to all possible combinations.

In order to interface with such electrical systems, a true understanding of the underlying physical communication layer is necessary. In the past, most modulation/demodulation systems were focused on a given encoding scheme and multiple techniques for recovering the digital information were proposed. Most of these methods focused on Fourier transform analysis where the information was contained in the spectral analysis of the sinusoidal signal functions. These systems (mostly specialized ASICS) are pre-programmed for specific classes of signals to be modulated/demodulated following a specific standard. Because of their specificity, the encoding mechanisms are based on the Fast Fourier Transform (FFT) for demodulation and the Inverse Fast Fourier Transform (IFFT) for modulation, both discrete versions of the Fourier Transform (FT).

Using the Fourier Transform (FT), typical time-domain signals can be represented as a sum of sinusoidal functions in the spectral domain. However, the number of functions needed to represent any arbitrary signal including those with discontinuities is infinite (Gibbs phenomenon). Therefore, the Fourier coefficients are also infinite. In order for a decoding mechanism to apply to both continuous and non-continuous signals, it is necessary to preserve the frequency information as well as the temporal information of the signal. Due to the large constellation of points involved in decoding higher-order modulation schemes, more templates are required to properly detect the amplitude and phase changes. Thus, higher-order modulation schemes can require an enormous number of templates to be used when attempting to decode a signal. This is obviously very resource intensive and may be impractical for resource constrained applications.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

To easily identify the discussion of any particular element or act, the most significant digit or digits in a reference number refer to the figure number in which that element is first introduced.

FIG. **1** illustrates an embodiment of a system for automatic electrical signal discovery and decoding **100**.

FIG. **2** illustrates a process for automatic electrical signal discovery and decoding **200**.

FIG. **3** illustrates an embodiment of a process for automatic electrical signal discovery and decoding **300**.

FIG. **4** illustrates an embodiment of a signal demodulator/modulator/modulator **400**.

FIG. **5** illustrates an embodiment of a signal demodulator/modulator/modulator **400**.

FIG. **6** illustrates an embodiment of automatic signal discovery and decoding **600**

FIG. **7** illustrates an example of decomposition and reconstruction wavelet and scaling functions **700**.

FIG. **8** illustrates constellation diagrams **800**.

FIG. **9** illustrates an apparatus **300** in accordance with one embodiment.

DETAILED DESCRIPTION

Glossary

"electric disturbance" in this context refers to a change in the voltage or current of an electrical signal;

"frequency filter" in this context refers to logic to filter out a frequency or frequencies;

"signal demodulator" in this context refers to a module which receives electrical signals and decodes them by applying a wavelet transform to the signal;

"signal filter window" in this context refers to logic to examine a portion of an electrical disturbance (signal), this logic may utilize a window function;

DESCRIPTION

This system and method allows the user to decode virtually any wired or wireless signal. The utilization of wavelet transforms and common-feature templates (templates) allow for the efficient analysis and decoding of signals which may not otherwise have been possible. The method described herein is applicable to many existing encoding mechanisms used between a remote control and a controlled device and allows for modulation and demodulation on both the software and hardware level. It is also possible to apply the method to future encoding mechanisms and can be understood as a flexible, dynamic and adaptive modem.

The system and method use a Wavelet Transform (WT) approach to modulate and demodulate unknown electrical communication signals. The principles of the Automatic Signal Discovery (ASD), modulation, decoding and subsequent encoding and WT noise reduction are well known in image processing and the same de-noising techniques can be used for electrical communication systems. However, it may be important to note that because the noise is distributed uniformly across all wavelet coefficients, a linear method will yield better signal data preservation. Corruption detection using the present method is possible but if the initial sampled data is corrupted, de-noising may yield degraded information during Automatic Signal Discovery.

For wireless systems, in order to improve the ASD, further signal processing and conditioning are required. A mathematical understanding on how the signal is propagating from the transmitter to the receiver, allows the receiving system to adapt to the transmission conditions. Further, minimizing or cancelling effect of group and phase delay between different frequency components is a typical pre-processing step done for wireless systems.

In the present system and method, the Automatic Signal Discovery process is based on a Continuous Wavelet Transform of a digitally modulated signal with an unknown modulation scheme. This signal is an input to a receiver (a device) generated from a remote control, or vice-versa. This CWT is then cross-correlated to a set of Wavelet Domain (WD) common-feature templates which may have previously been generated.

Generally, wavelets are purposefully crafted to have specific properties that make them useful for signal processing. The Wavelet Transform has been applied to identify a wide ranges of signals spanning multiple fields. Typically, using feature specific templates, the wavelet coefficients of a given signal are plotted on a histogram, and counting the number of peaks in the histogram allows discrimination between different encoding mechanisms such as the popular Phase-Shift Keying (PSK) and Frequency Shift Keying (FSK) modulation schemes. A special case of the Daubechies and Haar wavelet, have been used extensively to recover these wavelet coefficients.

It is well-known that for feature extraction of images and biometric information, the Biorthogonal family of wavelet exhibits the property of linear phase, which is needed for signal decomposition and reconstruction. By using two wavelets FIG. **7**, one for decomposition (on top) and the other for reconstruction (on bottom) instead of the same single one, properties suitable for ASD in the present system and method are derived.

Wavelets may be combined, using a "reverse, shift, multiply and integrate" technique called convolution, with portions of a known signal to extract information from the unknown signal. The Wavelet Transform (WT) provides a superior and flexible means of automatic modulation and demodulation system. With the WT, wavelet coefficients are attained by translating and dilating the time windows for a given resolution and are cross-correlated with the initial signal. With a WT, varied and vendor-specific non-stationary signals can be decomposed into classes of encoding to recover the digital information automatically. In addition, wavelets are localized in both time and frequency domains which allow for far more resolution than with any other methods.

For a person of ordinary skill in the art, the most intuitive type of wavelet transform to use may be the Discrete Wavelet Transform (DWT), as they may feel that the DWT is computationally more efficient at dealing with a finite set of data-points allowed by the use of filters. Despite the DWT's computational efficiency, however, it is generally unsuitable for current implementations of the system and method. Use of the DWT in this method causes skipped samples when cross-correlating the signals with any of the candidate wavelets and skipped samples when dilating candidate wavelets. Instead, a Continuous Wavelet Transform (CWT) function is used. A WT is a cross-correlation of a wavelet and a signal. Since most useful signals are gathered on a continuous time basis, the resulting wavelet is a form of a time-frequency approximation of a continuous-time (analog) signal. Another advantage in using a CWT is that it requires only the use of a single wavelet, instead of a wavelet and a scalet as with a DWT. Hence, a CWT is more suitable in this Automatic Signal Discovery (ASD) system and method as it preserves most of the signal content with wavelet coefficients at various levels of resolution, and provides a more complete representation of the convolution operation. On the other hand, a DWT is only suitable for de-noising and lossy compression of the original signal.

A hardware platform may be configured to automatically sense voltage and current levels, within a certain predetermined range, and then adapt its electronic circuitry to a given bandwidth to potentially detect a data stream and encoding mechanism. It is important to note that both the CWT and DWT are point-to-point discrete digital operations. They are both easy to compute with the use of a computer or specialized electronic hardware device (DSP, FPGA).

In one embodiment, the platform may be used for the discovery of physical encoding of communication data transferred between a commercial or household appliance and a remote control (external bus). The platform may interface with an existing analog electric system, and may assume that digital information might be encoded within this given analog system.

The system may passively monitor interconnection circuitry (between a given device and a its remote interface device). This communication link may be opened or closed (i.e. one of the device may or may not be connected to form a closed circuit). In the case of a closed circuit, the hardware platform connects as a parallel circuit, while minimizing the overall circuit impedance when connected to the bus. The electrical voltage (hence current) is quantized in an Analog to Digital Converter (ADC) and sampled at a given rate. The resulting digital information may be rendered as binary 1's and 0's. In all cases, the sampling period determines a time stamp for each sample and is stored. If the circuit is varying voltage or current on more than 1-wire (i.e. not ground, periodic or steady state level), a corresponding number of files per channel are generated which contain that particular channel ADC in the sampling order. Time information can be recovered by multiplying the sampling order to the sampling period.

The physical topologies may be based on different interfaces for both wired and wireless schemes, by way of example: bit-serial (1-, 2-, 3-, 4- and 5-wire), parallel (4-, 8-, 12- or 16-wire), and wireless radios. If the appliance and its remote control communicate wirelessly e.g. electromagnetic waves, optical, acoustics or mechanically, an appropriate transducer to correlate the carrier wave (which contains the digital information) to an electrical signal is used. Automatically recognized data streams are primarily serial asynchronous data transfer mechanisms based on actual binary data transfer modulated as shifting voltage, current and/or frequency levels.

Encoding schemes based on time modulation are also recognized if used as part of data transfer. Such time modulation may be based on a pulse width modulation (PWM), or pulse-duration modulation (PDM). In such cases, a direct correlation between the period and the clock basis of the signal yield a duty-cycle. The duty-cycle is transduced as binary numbers to represent the encoded data.

Digital Modulation Schemes

In a modern wired system, digital data may be modulated into an analog signal prior to transmission. It is fundamentally well understood how a wire carries electricity (using Ohm's Law, a signal can be encoded by modulating Voltage or Current). So, for the wired system encoding mechanisms discussed below, the modulation can use either voltage or current as the representative electromagnetic medium. Because current remains constant in any circuit loop (Kirchhoff s Law), current modulation is sometimes preferred over voltage. At any rate, the current can be deduced from the ensuing voltage as it passes through a known resistance.

It is known in the art that different encoding techniques are used depending on the application, and the medium used to transfer information. Often, the most cost-efficient method to arrive to good performance (signal-to-noise ratio) is sought. Each of the modulation scheme shown below carry a symbol from a family of $M=2^N$ symbols or bits.

The following encoding techniques (ASK, FSK, PSK, QAM) are well understood and widely used in the art and are used by way of example to explain the method for decoding both known and novel encoding mechanisms.

Amplitude Shift Keying (ASK) is a form of modulation which represents digital data as variations in the amplitude of a carrier wave:

$$s_i(t) = \begin{cases} A_i\sqrt{\frac{2E}{T}}\cos(2\pi f_c t),\ 0 \le t \le T \\ 0,\ \text{for all other } t \end{cases}$$

where i∈ℕ *i.e i=1, 2, 3, . . . A_i the signal amplitudes represent a vector of the binary symbols 1 or 0.

For BASK i=1, 2, hence 2 symbols encoding 0 and 1.

For a 4 ASK, 4 amplitudes encoding [00], [01], [10], [11] and so on . . .

Equation 1: ASK Time-Domain Equation

Frequency Shift Keying (FSK): In Frequency Shift Keying, the change in frequency define different digital symbols.

Equation 2

$$s_i(t) = \begin{cases} \sqrt{\frac{2E}{T}}\cos(2\pi f_c t),\ 0 \le t \le T \\ 0,\ \text{for all other } t \end{cases}$$

FSK time-domain equation

Phase Shift Keying (PSK): The phase of the carrier is discretely varied in relation either to a reference phase or to the phase of the immediately preceding signal element, in accordance with data being transmitted

$$s_i(t) = \begin{cases} A_i\sqrt{\frac{2E}{T}}\cos(2\pi f_c t + \beta),\ 0 \le t \le T \\ 0,\ \text{for all other } t \end{cases}$$

where β=**2**π/O (i−1) O denote order of the PSK signal.

When i=1, 2, 3, 4, the data symbol encoding [00], [01], [10], [11] respectively

Equation 3: Other M-PSK Signals

Quadratic Amplitude Modulation (QAM): is a modulation scheme where the amplitude of 2 waves of the same frequency, π/2 out-of-phase with each other are changed to represent the digital data:

$$s_i(t) = a_i\sqrt{\frac{2E}{T}}\cos(2\pi f_c t) + b_i\sqrt{\frac{2E}{T}}$$

where a_i and b_i represent the amplitude of in-phase and quadrature carriers.

Note that the carrier frequency remains constant.

Equation 4: M-QAM Signals

The template generation technique described below can be used to generate common-feature templates. It is important to note that a specific template is needed for any known digitally modulated signal. Because of PSK's simplicity over QAM, many wireless standards use it. The very successful QPSK IEEE 802.11b-1999 uses, for example, different PSK modulations for different data rates as required.

It is also used in RFID, Bluetooth and others systems. PSK and 8-PSK are widely used in satellite broadcasting and Standard Definition satellite television signals. More complex systems, are still use a combination of QPSK and 8PSK such as DVB-S2 systems. The signal representation shown earlier under wired systems also applies for wireless systems.

Template Library Generation

Automatic Signal Discovery and Decoding are attained by the use of Wavelet Domain templates, which are used in the cross-correlation step of the ASD. The template generation is based on the fact that all digitally modulated communication signals of interest are sinusoidal in nature. This is defined as:

$$s_i(t) = A_c\cos(2\pi f_c t)$$

where f_c is the carrier frequency and A_c the carrier amplitude of the signal $s_i(t)$*

Equation 5: A Digitally Modulated Communication Signal

To enable a template to recognize temporal shifts of the carrier signal, a phase shift is introduced to the equation above as:

$$t_i(t) = A_c\cos(2\pi f_c t + \phi),\ t_1 \le t \le t_2$$

where ϕ is the phase shift of the signal $t_i(t)$

Equation 6: A Template Model for Detecting Signals which Modulate the Carrier Frequency to Transfer Data

The above equation is the base model representing the decoding template in the time-domain. Using the signal definition as shown in eq. 1, 2, 3 and the template model shown in Equation 6, we can substitute the carrier frequency to compute, then store the cross-correlation values between them. The computation covers all the phase ranges −π to +π. The template generation method generates at least 3 separate templates for a given modulation technique. It is accomplished by modulating the cosine and sine carrier wave with:

σ=0 1st quadrant of the in-phase and quadrature (IQ)-plane, Modulation Template **1 610**

σ=π/2 2nd quadrant of IQ-plane, Modulation Template **2 614**

σ=5π/4 3rd quadrant of IQ-plane, Modulation Template **3 622**

The wavelet cross-correlation with the different version of Equation 6 represents mathematically a combination of the I and Q basis functions.

By way of example, Template **1** represents the coordinates along with the different metrics relating to the cross-correlation values. It further allows the identification of the encoding mechanism and the transmitted symbol encoded in the signal. The wavelet may be selected such that the confidence level (closest constellation point) is high.

In the case where the modulation scheme is contained between cosine and phase-shifts to a sine carrier frequency on the IQ plane,

$$\left(\cos\left(f(t) + \frac{\pi}{2}\right) = \sin f(t)\right)$$

only 2 templates (Template **1 618** and Template **2 620**) based on the cosine and sine coordinates are used. This is possible because the temporal shifts of a signal in the time-domain correspond to the translation variable in the wavelet-domain. Hence, phase-shifts in the time domain have a time-shifting effect in the wavelet-domain.

Template **1** **618** gives the I-axis coordinate, Template **2** **620** gives the Q-axis. This allows describing with Template **1** BPSK signals directly. With Template **1** and **2**, other M-ary PSK signals may be described. Template **1**, **2** and **3** are used to identify and retrieve an 8-PSK signal. When more than 8 symbols are encoded in a Phase-Shift-Keying modulation scheme, it is well known that the signal-to-noise ratio increases to a point that there is significant data loss (Shannon-Hartley Theorem). In this case, other defining coordinate bound templates are needed to fully describe and locate the multiple bits sent by symbols. This can be understood as the symbol points on the signal constellation being much closer together, hence more susceptible to noise and other corruptions. Typically, those amplitude based encoding schemes such as the amplitude-shift keying (ASK) or quadratic-amplitude-modulation (QAM) will require Template **1** and additional templates.

Since QAM signals encode 2 digital bit streams by modulating the amplitudes of 2 out of phase signals (by π/2), the additional templates for QAM are not encoding the carrier frequency. If the frequency is changed along with phase and amplitude, then those signals would smear making decoding impossible.

$$q_1(t)=A_1\cos(2\pi f_c t)+A_2\cos(2\pi f_c t),\ t_i\le t\le t_{i+1}$$

$$q_2(t)=B_1\cos(2\pi f_c t)+B_2\cos(2\pi f_c t),\ t_i\le t\le t_{i+1}$$

$$q_3(t)=C_1\cos(2\pi f_c t)+C_2\cos(2\pi f_c t),\ t_i\le t\le t_{i+1}$$

$$q_4(t)=D_1\cos(2\pi f_c t)+D_2\cos(2\pi f_c t),\ t_i\le t\le t_{i+1}$$

where A_1, A_2, B_1, B_2, C_1, C_2, D_1 and D_2 depend on the location of the template on the signal constellation. The i index is the i-th transmitted symbol in the QAM transmission

Equation 7: Base Templates for QAM Signals Describing Location in all 4 Quadrant of the IQ Plane

In this case, the additional templates based on Equation 7, by way of example, are used to describe shifting square signal constellations. That is, the QAM signal size expands to cover the square area to enfold all the possible data symbol locations. The signal cross-correlations with Template **1** and **2** give an indication in which initial quadrant the signal symbol lie. The additional templates are used to identify on that square plane which symbol is being transmitted. It is a direct result that when the cross-correlation of the received signal with Template **1** and **2** are the same, that the signal is a QAM signal. Cross-correlating with the quadrant specific templates would indicate the QAM signal. This methodology may also be used in cases where the QAM signal is not evenly spaced and would require generating templates which would account for the Euclidean distances (x,y) between the points. The auto-correlation and cross-correlation results with such signals would quickly indicate wether it was evenly spaced or not.

A major advantage of this method over currently used techniques based on unique feature templates, is that for higher-order modulation schemes the number of required templates increases dramatically. Due to the larger constellation of points, more templates are required to properly detect the amplitude and phase changes. For example, a 256-QAM would require 2^16=65536 templates when using a unique feature template method.

The system and method instead utilize common feature templates, which need at the very least 2 common feature templates and a maximum of 32 to fully represent all commonly used digital modulation schemes. The main benefit as opposed to the state-of-the art methods is a large reduction in computational complexity. Also, the WD cross-correlation data is much smaller and is effective in additional cross-correlation computations.

This method's use of the above common template library build for different carrier waves allows for numerical representation of all commonly used digital modulation schemes. As opposed to the state-of-the art methods, the disclosed method leads to reduced computational complexity and reliance on a single method to both recognize the modulation scheme and data stream decoding.

Additionally, the WD cross-correlation data is much smaller and is effective in the successive convolutions. The cross correlation values computed by the above template generation process are used for both the ASD and demodulation of the original signal.

CWT may be selected by computing the auto-correlation from a family of wavelets already in use in other techniques. The candidate wavelets are those producing a large magnitude changes in the CWT coefficients for our known modulation mechanisms. In, addition to the generated cross-correlation values, additional key information are computed. A statistical multi-level test of this data is obtained to further determine nuances in QAM modulation where the currently selected constellation quadrant may be expanding to widening constellation squares. This information is used in the ASD process.

$$Y_{ij}=\beta_{0j}+\beta_{1j}X_{ij}+e_{ij}$$

1-level regression

$$\beta_{0j}=\gamma_{00}+\gamma_{01}W_j+v_{0j}$$

$$\beta_{1j}=\gamma_{10}+v_{1j}$$

2-level regression

Equation 8: Statistical Categorization

Template 1		
Encoding System	Normalized Dynamic Range	Statistical Categorization
BASK	0 to 5	0
4-ASK	0 to 5	1
BFSK	0 to 5	0
4-FSK	0 to 5	0
BM	−5 to 5	0
QPSK	−5 to 5	2
8-PSK	−5 to 5	2
Al-QAM	−5 to 5	0
n-QAM (2^n, n = 2..4)	−5 to 5	3

Template 2		
Encoding System	Normalized Dynamic Range	Statistical Categorization
BASK	0 to 5	0
4-ASK	0 to 5	0
BFSK	0 to 5	0
4-FSK	0 to 5	0
BPSK	−5 to 5	0
QPSK3	−5 to 5	2
8-PSK	−5 to 5	2
n-QAM (2^n, n = 1..4)	−5 to 5	3

-continued

Template 3		
Encoding System	Normalized Dynamic Range	Statistical Categorization
BASK	0 to 5	0
4-ASK	0 to 5	1
BFSK	0 to 5	0
4-FSK	0 to 5	0
BPSK	−5 to 5	0
QPSK	−5 to 5	2
8-PSK	−5 to 5	2
n-QAM (2^n, n = 1..4)	−5 to 5	3

Sample Modulation Scheme Selection Table for Popular Modulation Schemes

Data Decoding

The ASD technique may only be useful for digitally encoded baseband data. In Equation 6, the phase variable represents the shifting of the templates within an encoded symbol base. Data decoding is dependent on the discovered encoding mechanism since the shifting requires any combination of the generated templates.

Again, using the previously obtained normalized cross-correlation values between the templates and now known modulated signal, an process may be developed to decode the baseband bit stream.

The basis for the decoding process is the knowledge of the modulation signal transition in time and the resulting cross-correlation at those exact locations. The set of cross-correlation values obtained with our specially created templates, shows an absolute level of correlation confidence at a specific location for "1-encoded" bit transitioning to the next as shown on the constellation diagram in FIG. **8**. The process also uses a set of threshold metrics to uniquely identify the concerned bit transition from a "1-encoded" value to the next. The bit transition grouping is dependent on the discovered encoding mechanism since some communication modulation encode more than 2 bits at any given time. For such systems, the process will use of additional cross-correlation values from the additional template to increase the number of recovered bits at each time location.

To recover the bit or bits encoded in a signal at time t (depending on the ASD output), we judiciously chose the following:

Modulation Template **1** maps to the constellation points on the I-axis and when M=1 (M number of symbols in signal), it is sufficient to recover the [0] or [1] (see in Word doc, template decoding, FIG. **8**—BPSK). The choice filter is as simple as comparing the cross-correlation value from Modulation Template **1** to either be positive or negative to chose [0] and [1] respectively.

Modulation Template **2** maps to the constellation points on the I-axis and since it is a sine axis, the WD cross-correlation values need Modulation Template **1** for completing the number of symbols for when M=2, i.e, n=2^2=4-bits ([00], [01], [10], [11]) (QPSK **804**). The choice filter follows the same logic as earlier. That is, if the cross-correlation value of Template **1** is positive and Template **2** is 0, then the signal recovered is [00], if the cross-correlation value of Template **1** is 0 and Template **2** is positive, then [01], if Template **1** value is negative, Template **2** is 0, then [10], finally if the value of Template **1** is 0 and Template **2** is negative then [11]

Modulation Template **3** maps to an even greater number of constellation points on the IQ-plane and require the use of the other 2 template. The choice filter for decoding follow a similar logic as earlier. The bits recovered for this operation where the number of symbols for M=3, is n=2^3=8-bit ([000] to [111]).

Finally when additional modulation template n are used, the same decoding logic applies using Template **1** as a base quadrant in the I-Q plane to search to the QAM constellation points matching the coordinate given by the cross-correlation. It may be faster to use auto-correlation with these additional templates and cross-correlate the value of Template **1** and Template **2** to compute the distance (Euclidean distance) between the points. These distances give a threshold indication used in the process to recover bits which lie inexactly in the constellation quadrant space.

Pseudocode for Automatic Signal Modulation Discovery (FIG. **6**)

1—Acquire signal from Analog to Digital System with unknown modulation scheme (In most cases, the sample rate must satisfy the Nyquist critera. In the present process, data is oversampled by a larger factor)

2—Apply a Continuous Wavelet Transform (CWT) (in our case, we use the biorthogonal rbio1.3 wavelet)

3—Do n=1 to 3, select template(n) from the template library

4—Cross-correlate signal acquired in 1—with template (n) and obtain cross-correlation values

5—Normalize the cross-correlation values to have a dynamic range from −5 to 5 by taking the ratio between largest and smallest values (fixed dynamic range) and scaling.

6—Repeat 3—until all templates' cross-correlation are computed.

7—If the normalized cross-correlation values with Template **2** are all 0, or only positive, or only negative, then do 7-b, else do 7-a

7-a Based on the algebraic sign of the cross-correlation result, select the quadrant specific templates

7-b Use a random model intercept to compute the Y(i,j) where i and j are the templates taken together such that (i,j)=(1,2), (1,3) and (3,2) to determine if there exists is zero-valued (all 3 Y(i,j)=[0]), single level (2 of the 3 combination is 0) or multi-level relationship.

7-c Assign 0 for zero-valued, 1 for single valued, 2 for multi-level relationship.

8—Detect encoding mechanism based on classification table.

It is important to note that for classifying binary higher-order modulation schemes, additional templates are required following the same processes for detection and template generation. In step 7, cross correlation may not be sufficient to properly classify densely populated signals. Further, these higher order encoding mechanism are more susceptible to noise corruption and the sampling hardware needs to be adjusted.

General Decoding Process

1—Initialize a binary code string S which is ordered from left to right where the most significant bit is right-most.

2—Using the ASD results, the list of used templates, their respective cross-correlation data from the discovered signal

3—If ASD detected an M=1 modulation scheme (BPSK), then

3-a For each cross-correlation value from Template **1** (from ASD), use the BPSK decision filter to recover bits, add to S, do until all Template **1** cross correlation values have been filtered. Goto end

4—If ASD detected an M=2 modulation scheme (QPSK), then

4-a For each cross-correlation value from Template **1** and Template **2**, use the QPSK decision filter to recover bits, add to S, do until all Template **1**, **2** cross correlation values have been filtered. Goto end

5—If ASD detected an M=3 modulation scheme (8-PSK), then

5-a For each cross-correlation value from Template **1**,**2**,**3** use the 8-PSK decision filter to recover bits, add to S, do until all Template **1**,**2**,**3** cross correlation values have been filtered. Goto end

6—If ASD detected a 4-FSK, a total of 5 Templates are needed, Template **1**, **4** and **5**. Do,

6-a Use filter for M=1 (BPSK), for each cross-correlation of signal with Template **1**, do

6-b If filter output is 0, then, if each cross correlation of Template **4** and **5** are both 0, add to S, [00], else

6-c Use filter following process given above (i.e., if T4>0 and T5=0, recover bit [01] etc. . . .), add to S

6-c increment all indexes in cross correlation

6-d goto 6.

7—If ASD detected an BASK signal, for each cross correlation values with Template **1**, use filter of 3—(i.e., if value=0, [0] and if value>0, [1]), add [0] or [1] to S, goto end

8—If ASD detected an M-ASK (now, data to be recovered is in the WD cross correlation values dynamic range)

8-a for each cross correlation dynamic range obtained with Template **1**, create 4 range from 0 to the 5 and for each value falling in range 0, assign [00], range 1 [01], range 2 [10] and range 3 [11], add to S, next, goto end

9—If ASD detected M-QAM signal, additional templates where used (templates follow eq.9)

9-a Determine IQ-plane from Template **1** and **2** cross-correlation values

9-b Using cross correlation data of template **3**, use filter as in 3-

9-c For each auto-correlation values with largest template N (defining the widest square)

9-d For i=0 to M/4−1 do,

9-e Assign Threshold x(N−i) for Template (N−i) corresponding to symbol position i by computing the auto-correlation of each Template(N−i), next

9-f Filter according to: compare each cross-correlation from 9b and assign bit for symbol (number of bits (M/4), next, write bits into S

10—End.

DRAWINGS

The system for automatic electrical signal discovery and decoding **100** comprises an em signal transducer **102**, a signal monitor **104**, a signal monitor **110**, a parallel bus **112**, a signal switch **106**, a signal demodulator/modulator/modulator **400**, a filter buffer **114** (deleted), and a channel code file **108**.

The system for automatic electrical signal discovery and decoding **100** may be operated in accordance with the processes described in process for automatic electrical signal discovery and decoding **200**, and process for automatic electrical signal discovery and decoding **300**.

The signal monitor **104** and may be electromagnetic signal monitors.

Automatic Signal Modulation Discovery: This is the core system use to select the appropriate matching templates for family of encoding techniques. The ASD is also used in the feedback mechanism to generate the appropriate signal to be transmitted.

Demodulator/Modulator: The demodulator/modulator are the secondary core system in the ASD

Referring to FIG. **2**, in block **202**, a process for automatic electrical signal discovery and decoding **200** detects electrical disturbances with a signal monitor and closes a signal switch to transmit the electrical disturbances to a signal demodulator/modulator. A signal filter window is configured on a frequency filter to maximize time domain signal resolution (block **204**). Signal processing is applied to transform the electrical disturbances into a demodulated signal with the signal demodulator/modulator (block **206**). The electrical disturbances are transmitted to the frequency filter to filter out a captured waveform that is captured by the signal filter window (block **208**). The signal filter window is configured to increase frequency domain signal resolution (block **210**). The signal processing is repeated until a maximal frequency resolution is attained (block **212**). The signal templates from a signal template library are applied to the captured waveform to calculating a template relationship value for each of the signal templates utilized (block **214**). The template relationship value is normalized (block **216**). A selector is operated with the template relationship value to select decoding templates (block **218**). A selector is operated with the template relationship value, and the decoding templates to select a modulation scheme from a classification table (block **220**).

Process for automatic electrical signal discovery and decoding **200** continues in process for automatic electrical signal discovery and decoding **300**;

In block **302** process for automatic electrical signal discovery and decoding **300** decodes the electrical disturbances. A binary code string is initialized (block **304**). The template relationship value, decoding templates and modulation scheme are applied to filter the electrical disturbances into decoded bits (block **306**). The decoded bits are written to the binary code string (block **308**). The binary code string is written to the channel code file (block **310**).

This process provides more balanced resolution between In addition, this process is an improvement by requiring operational complexity closer to constant time than current processes, thereby reducing the usage of processor resources and thereby increasing system efficiency.

This process increases system efficiency and resource in several ways. By utilizing common feature templates, number of possible templates needed to demodulate higher-order signals is greatly reduced. For example, demodulation of a 256-QAM would require 2^{16} (65,536) templates when using the unique feature templates, while the instant method uses a maximum of 32 to represent all commonly used digital modulation schemes. This results in a substantial reduction in computational complexity.

In addition, the Wavelet Domain cross-correlation data is much smaller and is effective in the successive convolutions.

This methods use of the above common template library build for different carrier waves allows for numerical representation of all commonly used digital modulation schemes.

The signal demodulator/modulator/modulator **400** comprises analog signal **426**, Resistor **422**, reference signal voltage **402**, waveform electrical data **408**, voltage comparator **404**, voltage down-shift **406**, analog to digital **410**, waveform memory capture **412**, CWT **414**, wavelet based automatic signal demodulation **416**, CWT coefficients **418**, and binary code string **420**.

Analog signal **426** is routed through Resistor **422** and compared to reference signal voltage **402** and the waveform electrical data **408** is retrieved. A voltage down-shift **406** is then applied and the analog signal **426** is then converted to digital via analog to digital **410**, logged in waveform memory capture **412** and CWT **414** is applied to process analog signal **426** via wavelet based automatic signal demodulation **416**. CWT coefficients **418** and **320** are retrieved from the wavelet based automatic signal demodulation **416**.

The signal demodulator/modulator/modulator **400** comprises generated (baseband) data **526**, Initial Data signal processor **502**, Modulator **504**, Radio Processing (for wireless) Digital to Analog (for wired) **506**, Transmitter/Receiver (wired/wireless) **514**, Radio Processing (for wireless) Analog to Digital (for wired) **510**, **412**, noise reduction **532**, channel equalization and estimation (for wireless system) **530**, automatic signal discovery **524**, automatic decoding system **534**, recovered (baseband) data **518**, Reconstruction wavelet function ψ **512**, wavelet based automatic signal demodulation **416** and decomposition wavelet function ψ **508**.

The automatic signal discovery and decoding **600** comprises automatic signal discovery **524**, an automatic decoding system **534**, Radio Processing (for wireless) Analog to Digital (for wired) **510**, recovered (baseband) data **518**, Automatic Signal Detection **608**, Modulation Template **1** **610**, Modulation Template **2** **614**, Modulation Template **3** **622**, Modulation Template n **616**, Decoding using Templates **612**, Template **1** **618**, Template **2** **620**, cross-correlation **626**, and auto-correlator and comparator **602**.

FIG. **9** illustrates several components of an exemplary apparatus **900** in accordance with one embodiment. In various embodiments, apparatus **900** may include a desktop PC, server, workstation, mobile phone, laptop, tablet, set-top box, appliance, or other computing device that is capable of performing operations such as those described herein. In some embodiments, apparatus **900** may include many more components than those shown in FIG. **9**. However, it is not necessary that all of these generally conventional components be shown in order to disclose an illustrative embodiment. Collectively, the various tangible components or a subset of the tangible components may be referred to herein as "logic" configured or adapted in a particular way, for example as logic configured or adapted with particular software or firmware.

In various embodiments, apparatus **900** may comprise one or more physical and/or logical devices that collectively provide the functionalities described herein. In some embodiments, apparatus **900** may comprise one or more replicated and/or distributed physical or logical devices.

In some embodiments, apparatus **900** may comprise one or more computing resources provisioned from a "cloud computing" provider, for example, Amazon Elastic Compute Cloud ("Amazon EC2"), provided by Amazon.com, Inc. of Seattle, Wash.; Sun Cloud Compute Utility, provided by Sun Microsystems, Inc. of Santa Clara, Calif.; Windows Azure, provided by Microsoft Corporation of Redmond, Wash., and the like.

Apparatus **900** includes a bus **902** interconnecting several components including a network interface **908**, a display **906**, a central processing unit **910**, and a memory **904**.

Memory **904** generally comprises a random access memory ("RAM") and permanent non-transitory mass storage device, such as a hard disk drive or solid-state drive. Memory **904** stores an operating system **912**.

These and other software components may be loaded into memory **904** of apparatus **900** using a drive mechanism (not shown) associated with a non-transitory computer-readable medium **916**, such as a floppy disc, tape, DVD/CD-ROM drive, memory card, or the like.

Memory **904** also includes database **914**. In some embodiments, server **200** (deleted) may communicate with database **914** via network interface **908**, a storage area network ("SAN"), a high-speed serial bus, and/or via the other suitable communication technology.

In some embodiments, database **914** may comprise one or more storage resources provisioned from a "cloud storage" provider, for example, Amazon Simple Storage Service ("Amazon S3"), provided by Amazon.com, Inc. of Seattle, Wash., Google Cloud Storage, provided by Google, Inc. of Mountain View, Calif., and the like.

References to "one embodiment" or "an embodiment" do not necessarily refer to the same embodiment, although they may. Unless the context clearly requires otherwise, throughout the description and the claims, the words "comprise," "comprising," and the like are to be construed in an inclusive sense as opposed to an exclusive or exhaustive sense; that is to say, in the sense of "including, but not limited to." Words using the singular or plural number also include the plural or singular number respectively, unless expressly limited to a single one or multiple ones. Additionally, the words "herein," "above," "below" and words of similar import, when used in this application, refer to this application as a whole and not to any particular portions of this application. When the claims use the word "or" in reference to a list of two or more items, that word covers all of the following interpretations of the word: any of the items in the list, all of the items in the list and any combination of the items in the list, unless expressly limited to one or the other.

"Logic" refers to machine memory circuits, non transitory machine readable media, and/or circuitry which by way of its material and/or material-energy configuration comprises control and/or procedural signals, and/or settings and values (such as resistance, impedance, capacitance, inductance, current/voltage ratings, etc.), that may be applied to influence the operation of a device. Magnetic media, electronic circuits, electrical and optical memory (both volatile and nonvolatile), and firmware are examples of logic. Logic specifically excludes pure signals or software per se (however does not exclude machine memories comprising software and thereby forming configurations of matter).

Those skilled in the art will appreciate that logic may be distributed throughout one or more devices, and/or may be comprised of combinations memory, media, processing circuits and controllers, other circuits, and so on. Therefore, in the interest of clarity and correctness logic may not always be distinctly illustrated in drawings of devices and systems, although it is inherently present therein.

The techniques and procedures described herein may be implemented via logic distributed in one or more computing devices. The particular distribution and choice of logic will vary according to implementation.

Those having skill in the art will appreciate that there are various logic implementations by which processes and/or systems described herein can be effected (e.g., hardware, software, and/or firmware), and that the preferred vehicle will vary with the context in which the processes are deployed. "Software" refers to logic that may be readily readapted to different purposes (e.g. read/write volatile or nonvolatile memory or media). "Firmware" refers to logic embodied as read-only memories and/or media. Hardware refers to logic embodied as analog and/or digital circuits. If an implementer determines that speed and accuracy are paramount, the implementer may opt for a hardware and/or firmware vehicle; alternatively, if flexibility is paramount, the implementer may opt for a solely software implementation; or, yet again alternatively, the implementer may opt for some combination of hardware, software, and/or firmware. Hence, there are several possible vehicles by which the processes described herein may be effected, none of which is inherently superior to the other in that any vehicle to be utilized is a choice dependent upon the context in which the vehicle will be deployed and the specific concerns (e.g., speed, flexibility, or predictability) of the implementer, any of which may vary. Those skilled in the art will recognize that optical aspects of implementations may involve optically-oriented hardware, software, and or firmware.

The foregoing detailed description has set forth various embodiments of the devices and/or processes via the use of block diagrams, flowcharts, and/or examples. Insofar as such block diagrams, flowcharts, and/or examples contain one or more functions and/or operations, it will be understood as notorious by those within the art that each function and/or operation within such block diagrams, flowcharts, or examples can be implemented, individually and/or collectively, by a wide range of hardware, software, firmware, or virtually any combination thereof. Several portions of the subject matter described herein may be implemented via Application Specific Integrated Circuits (ASICs), Field Programmable Gate Arrays (FPGAs), digital signal processors (DSPs), or other integrated formats. However, those skilled in the art will recognize that some aspects of the embodiments disclosed herein, in whole or in part, can be equivalently implemented in standard integrated circuits, as one or more computer programs running on one or more computers (e.g., as one or more programs running on one or more computer systems), as one or more programs running on one or more processors (e.g., as one or more programs running on one or more microprocessors), as firmware, or as virtually any combination thereof, and that designing the circuitry and/or writing the code for the software and/or firmware would be well within the skill of one of skill in the art in light of this disclosure. In addition, those skilled in the art will appreciate that the mechanisms of the subject matter described herein are capable of being distributed as a program product in a variety of forms, and that an illustrative embodiment of the subject matter described herein applies equally regardless of the particular type of signal bearing media used to actually carry out the distribution. Examples of a signal bearing media include, but are not limited to, the following: recordable type media such as floppy disks, hard disk drives, CD ROMs, digital tape, flash drives, SD cards, solid state fixed or removable storage, and computer memory.

In a general sense, those skilled in the art will recognize that the various aspects described herein which can be implemented, individually and/or collectively, by a wide range of hardware, software, firmware, or any combination thereof can be viewed as being composed of various types of "circuitry." Consequently, as used herein "circuitry" includes, but is not limited to, electrical circuitry having at least one discrete electrical circuit, electrical circuitry having at least one integrated circuit, electrical circuitry having at least one application specific integrated circuit, circuitry forming a general purpose computing device configured by a computer program (e.g., a general purpose computer configured by a computer program which at least partially carries out processes and/or devices described herein, or a microprocessor configured by a computer program which at least partially carries out processes and/or devices described herein), circuitry forming a memory device (e.g., forms of random access memory), and/or circuitry forming a communications device (e.g., a modem, communications switch, or optical-electrical equipment).

Those skilled in the art will recognize that it is common within the art to describe devices and/or processes in the fashion set forth herein, and thereafter use standard engineering practices to integrate such described devices and/or processes into larger systems. That is, at least a portion of the devices and/or processes described herein can be integrated into a network processing system via a reasonable amount of experimentation.

What is claimed is:

1. A method for automatic electrical signal discovery and decoding comprising:

detecting electrical disturbances with a signal monitor and closing a signal switch to transmit the electrical disturbances to a signal demodulator;

configuring a signal filter window on a frequency filter to maximize time domain signal resolution;

applying signal processing to transform the electrical disturbances into a demodulated signal with the signal demodulator, the signal processing comprising:

transmitting the electrical disturbances to the frequency filter to filter out a captured waveform that is captured by the signal filter window;

configuring the signal filter window to increase frequency domain signal resolution;

repeating the signal processing until a maximal frequency resolution is attained;

applying signal templates from a signal template library to the captured waveform to calculating a template relationship value for each of the signal templates utilized; normalizing the template relationship value;

operating a first selector with the template relationship value to select decoding templates; and

operating a second selector with the template relationship value and the decoding templates to select a modulation scheme from a classification table; and

decoding the electrical disturbances comprising:

initializing a binary code string;

applying the template relationship value, the decoding templates and the modulation scheme to filter the electrical disturbances into decoded bits;

writing the decoded bits to the binary code string; and

writing the binary code string to a channel code file.

2. The method of claim **1** wherein calculating the template relationship value further comprises computing a cross-correlation value attained from cross-correlating the captured waveform and the signal templates.

3. The method of claim **1**, wherein detecting the electrical disturbances with the signal monitor and closing the signal switch to transmit the signal to the signal demodulator comprises:

detecting an electromagnetic disturbance with an electromagnetic signal monitor and transforming electromagnetic disturbance into an electrical signal with an electromagnetic signal transducer.

4. The method of claim **1**, wherein the signal monitor detects electrical disturbances on multiple conductive lines in a parallel bus.

5. The method of claim **1**, wherein the electrical disturbances are fluctuations in voltage or current.

6. The method of claim **1**, wherein the template relationship value comprises a cross-correlation value.

7. The method of claim **1**, wherein operating the selector with the template relationship value to select the decoding templates further comprises utilizing an algebraic sign of the template relationship value to select the templates.

8. The method of claim **1**, wherein operating the first selector with the template relationship value to select the decoding templates further comprises using a random model intercept to determine if there exists a zero-valued, single level or multi-level relationship.

9. A computing apparatus, the computing apparatus comprising:

a processor; and a memory storing instructions that, when executed by the processor, configure the apparatus to:

detect electrical disturbances with a signal monitor and closing a signal switch to transmit the electrical disturbances to a signal demodulator;

configure a signal filter window on a frequency filter to maximize time domain signal resolution; apply signal processing to transform the electrical disturbances into a demodulated signal with the signal demodulator, the signal processing comprising:

transmit the electrical disturbances to the frequency filter to filter out a captured waveform that is captured by the signal filter window;

configure the signal filter window to increase frequency domain signal resolution;

repeat the signal processing until a maximal frequency resolution is attained;

apply signal templates from a signal template library to the captured waveform to calculating a template relationship value for each of the signal templates utilized;

normalize the template relationship value;

operate a first selector with the template relationship value to select decoding templates; and

operate a second selector with the template relationship value and the decoding templates to select a modulation scheme from a classification table; and decode the electrical disturbances comprising:

initialize a binary code string;

apply the template relationship value, the decoding templates and the modulation scheme to filter the electrical disturbances into decoded bits;

write the decoded bits to the binary code string; and

write the binary code string to a channel code file.

10. The computing apparatus of claim **9** wherein calculating the template relationship value further comprises compute a cross-correlation value attained from cross-correlating the captured waveform and the signal templates.

11. The computing apparatus of claim **9**, wherein detecting the electrical disturbances with the signal monitor and closing the signal switch to transmit the signal to the signal demodulator comprises:

an electromagnetic signal monitor detecting an electromagnetic disturbance and an electromagnetic signal transducer transforming the electromagnetic disturbance into an electrical signal.

12. The computing apparatus of claim **9**, wherein the signal monitor detects electrical disturbances on multiple conductive lines in a parallel bus.

13. The computing apparatus of claim **9**, wherein the electrical disturbances are fluctuations in voltage or current.

14. The computing apparatus of claim **9**, wherein the template relationship value comprises a cross-correlation value.

15. The computing apparatus of claim **9**, wherein operating the first selector with the template relationship value to select the decoding templates further comprises utilizing an algebraic sign of the template relationship value to select the templates.

16. The computing apparatus of claim **9**, wherein operating the first selector with the template relationship value to select the decoding templates further comprises using a random model intercept to determine if there exists a zero-valued, single level or multi-level relationship.

* * * * *

File name: US9774480B1.docx

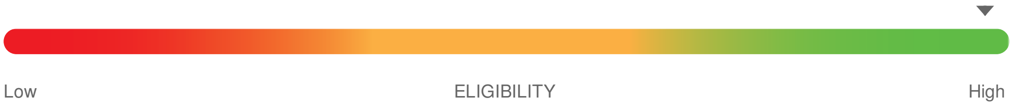
Art Unit Predictions

Statistics for the five most-likely results, in decreasing order

Art Unit	Allowance rate	Pendency (months)	Avg. no. of Office actions	% granted with appeal
2634	93%	22	1.4	4%
2631	90%	23	1.5	4%
2632	91%	22	1.5	4%
2633	95%	20	1.3	3%
2637	93%	25	1.3	3%

§101

Eligibility based on similarity to claims rejected under 101 for abstraction



Words related to low eligibility:

window	*maximize*	*value*	*discovery*
closing	*relationship*	*templates*	*transform*
maximal	*selector*	*scheme*	*automatic*
calculating	*classification*		

§102 §103

Novelty based on most-similar pieces of art

Related patent documents

9774480	8447620	20130096930	9043215
20170133023	20110238425	6311158	9008811
20170256267	6138093		

§112

Clarity issues based on language defects in the application

0	0	0	0
Antecedent basis issues	Figure reference issues	Unsupported claim terms	Claim order and format issues

Antecedent-basis issues

RoboReview™ found no antecedent-basis issues

Claim support issues

RoboReview™ found no claim support issues

Claim number and format issues

RoboReview™ found no claim number and format issues

Parts list

RoboReview™ found no issues with numbered parts

TurboPatent Sample Patent

CONTEXT- AND ACTIVITY-AWARE CONTENT SELECTION

US 9,727,561

August 8, 2017

US009727561B1

(12) United States Patent
Ko et al.

(10) **Patent No.: US 9,727,561 B1**
(45) **Date of Patent: Aug. 8, 2017**

(54) **CONTEXT- AND ACTIVITY-AWARE CONTENT SELECTION**

(71) Applicant: **Answerdash Inc.**, Seattle, WA (US)

(72) Inventors: **Andrew Ko**, Seattle, WA (US); **Victor Medina**, Seattle, WA (US)

(73) Assignee: **Answerdash Inc.**, Seattle, WA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **15/066,974**

(22) Filed: **Mar. 10, 2016**

Related U.S. Application Data

(60) Provisional application No. 62/131,218, filed on Mar. 10, 2015.

(51) **Int. Cl.**
G06F 17/30 (2006.01)
G06F 17/21 (2006.01)

(52) **U.S. Cl.**
CPC ***G06F 17/30011*** (2013.01); ***G06F 17/212*** (2013.01); ***G06F 17/3053*** (2013.01)

(58) **Field of Classification Search**
CPC G06F 17/30011; G06F 17/3053
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

2002/0091993 A1* 7/2002 Walley G06F 9/4446 717/120
2016/0239487 A1* 8/2016 Potharaju G06F 17/3053

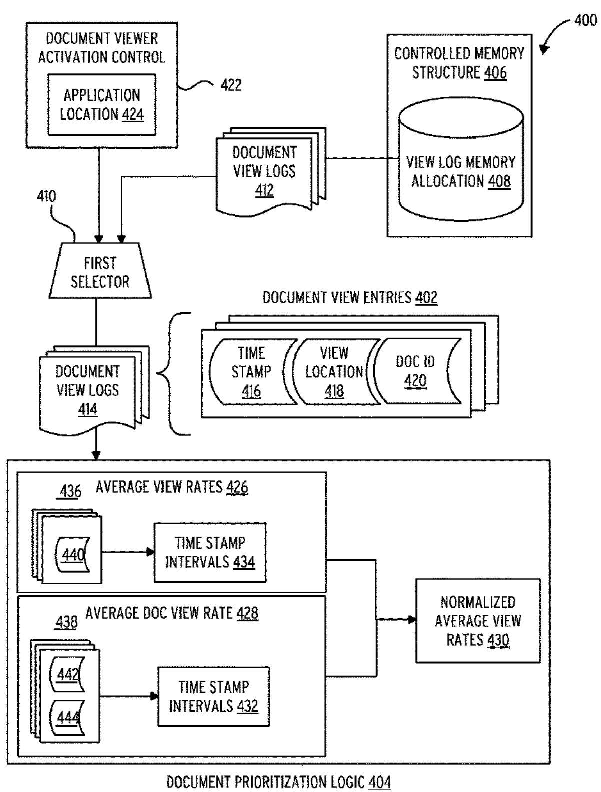
* cited by examiner

Primary Examiner — Asher Kells
(74) *Attorney, Agent, or Firm* — FSP LLC

(57) **ABSTRACT**

A method of contextual activity awareness content selection operates a switch to release a document viewer activation control to a first selector, operate the first selector to select the application location for the application interface, selects document view logs, associated with application location, operates document prioritization logic to calculates average document view rate, calculates an average view rate for each document at the application location, normalizes, weights, and rank average view rates document relevance set, operates a second selector to select matching documents and release to the document viewer, configures the document viewer to prioritize rendering based on document relevance set, operates the switch to release a document view signal to a compiler, in response to a document viewing in the document viewer, and operates the compiler to generate a new document view entry for the document viewing.

15 Claims, 16 Drawing Sheets



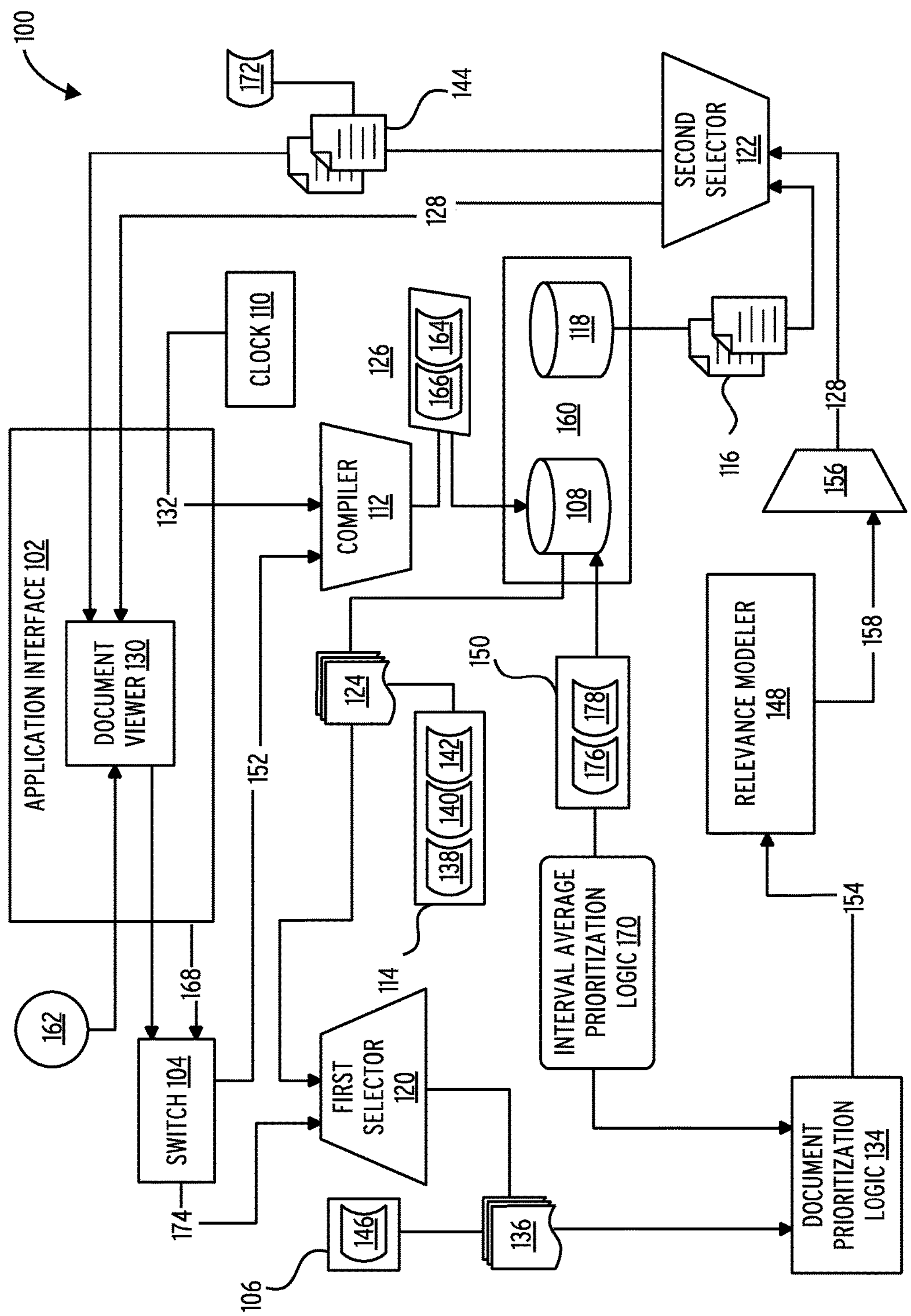

100
APPLICATION INTERFACE 102
DOCUMENT VIEWER 130
SWITCH 104
162
168
174
152
132
CLOCK 110
COMPILER 112
126
166
164
160
108
118
116
128
144
172
SECOND SELECTOR 122
124
138
140
142
114
FIRST SELECTOR 120
106
146
136
150
176
178
INTERVAL AVERAGE PRIORITIZATION LOGIC 170
DOCUMENT PRIORITIZATION LOGIC 134
RELEVANCE MODELER 148
154
158
156

FIG. 1

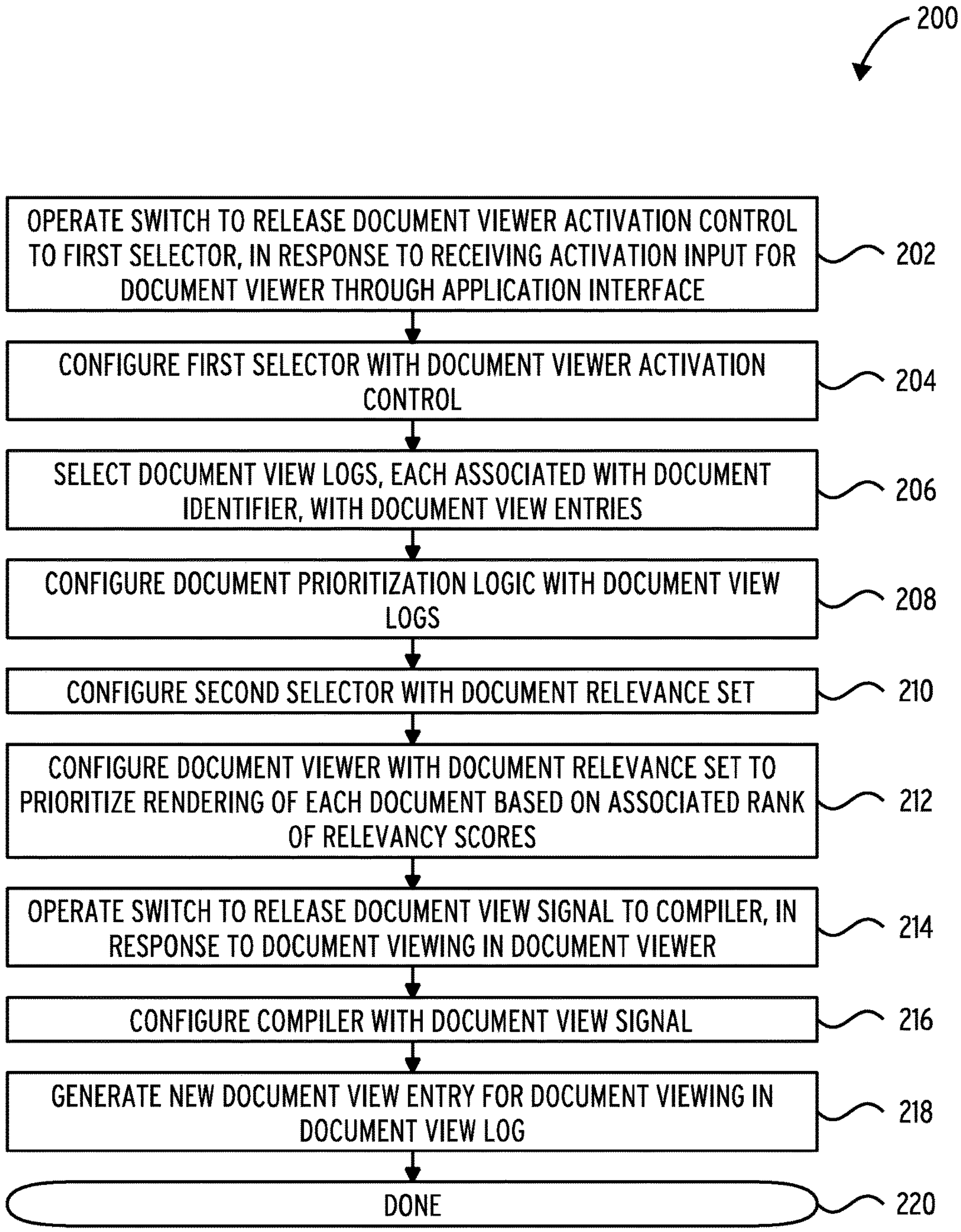

200
OPERATE SWITCH TO RELEASE DOCUMENT VIEWER ACTIVATION CONTROL TO FIRST SELECTOR, IN RESPONSE TO RECEIVING ACTIVATION INPUT FOR DOCUMENT VIEWER THROUGH APPLICATION INTERFACE
202
CONFIGURE FIRST SELECTOR WITH DOCUMENT VIEWER ACTIVATION CONTROL
204
SELECT DOCUMENT VIEW LOGS, EACH ASSOCIATED WITH DOCUMENT IDENTIFIER, WITH DOCUMENT VIEW ENTRIES
206
CONFIGURE DOCUMENT PRIORITIZATION LOGIC WITH DOCUMENT VIEW LOGS
208
CONFIGURE SECOND SELECTOR WITH DOCUMENT RELEVANCE SET
210
CONFIGURE DOCUMENT VIEWER WITH DOCUMENT RELEVANCE SET TO PRIORITIZE RENDERING OF EACH DOCUMENT BASED ON ASSOCIATED RANK OF RELEVANCY SCORES
212
OPERATE SWITCH TO RELEASE DOCUMENT VIEW SIGNAL TO COMPILER, IN RESPONSE TO DOCUMENT VIEWING IN DOCUMENT VIEWER
214
CONFIGURE COMPILER WITH DOCUMENT VIEW SIGNAL
216
GENERATE NEW DOCUMENT VIEW ENTRY FOR DOCUMENT VIEWING IN DOCUMENT VIEW LOG
218
DONE
220

FIG. 2

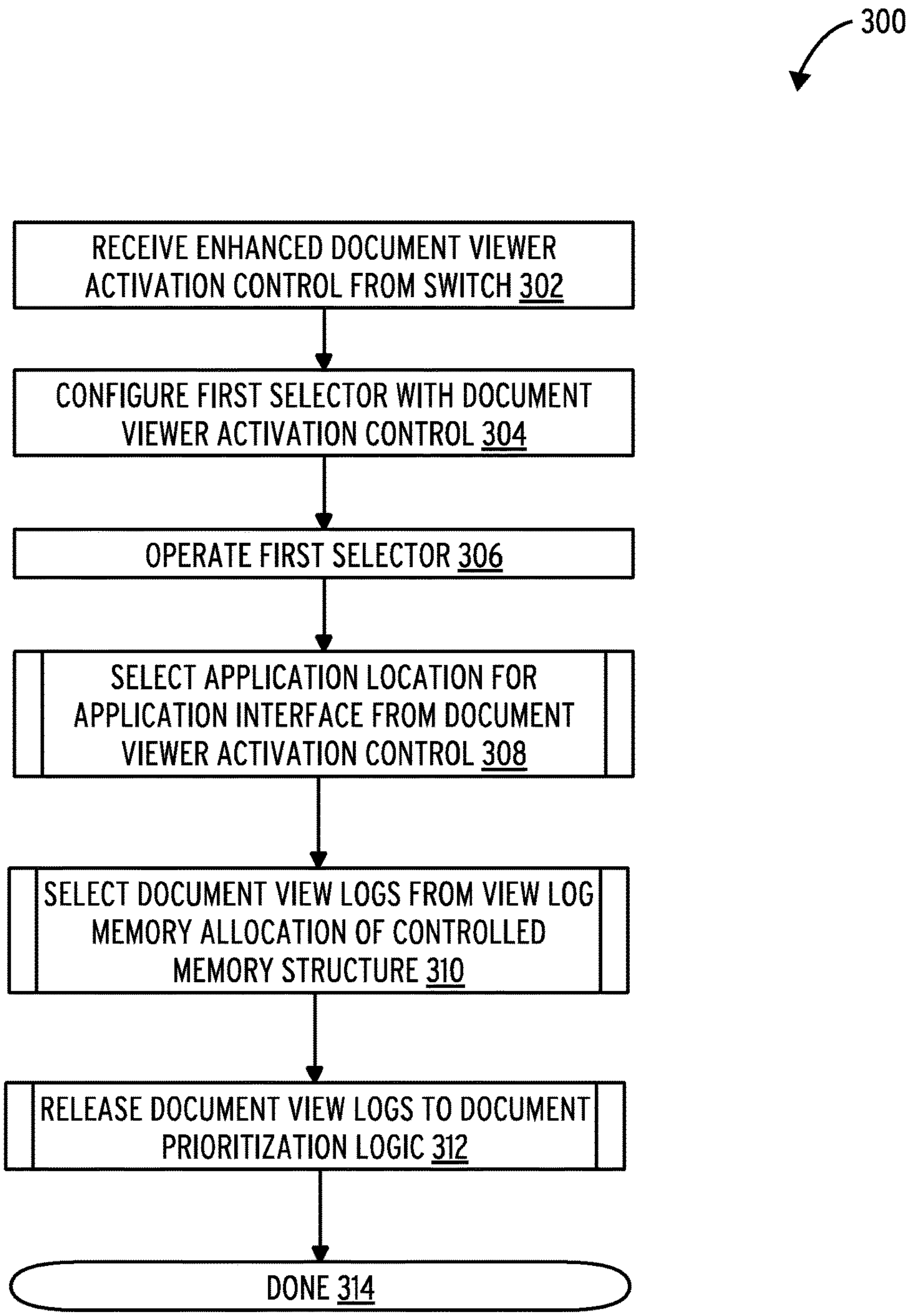

300
RECEIVE ENHANCED DOCUMENT VIEWER ACTIVATION CONTROL FROM SWITCH 302
CONFIGURE FIRST SELECTOR WITH DOCUMENT VIEWER ACTIVATION CONTROL 304
OPERATE FIRST SELECTOR 306
SELECT APPLICATION LOCATION FOR APPLICATION INTERFACE FROM DOCUMENT VIEWER ACTIVATION CONTROL 308
SELECT DOCUMENT VIEW LOGS FROM VIEW LOG MEMORY ALLOCATION OF CONTROLLED MEMORY STRUCTURE 310
RELEASE DOCUMENT VIEW LOGS TO DOCUMENT PRIORITIZATION LOGIC 312
DONE 314

FIG. 3

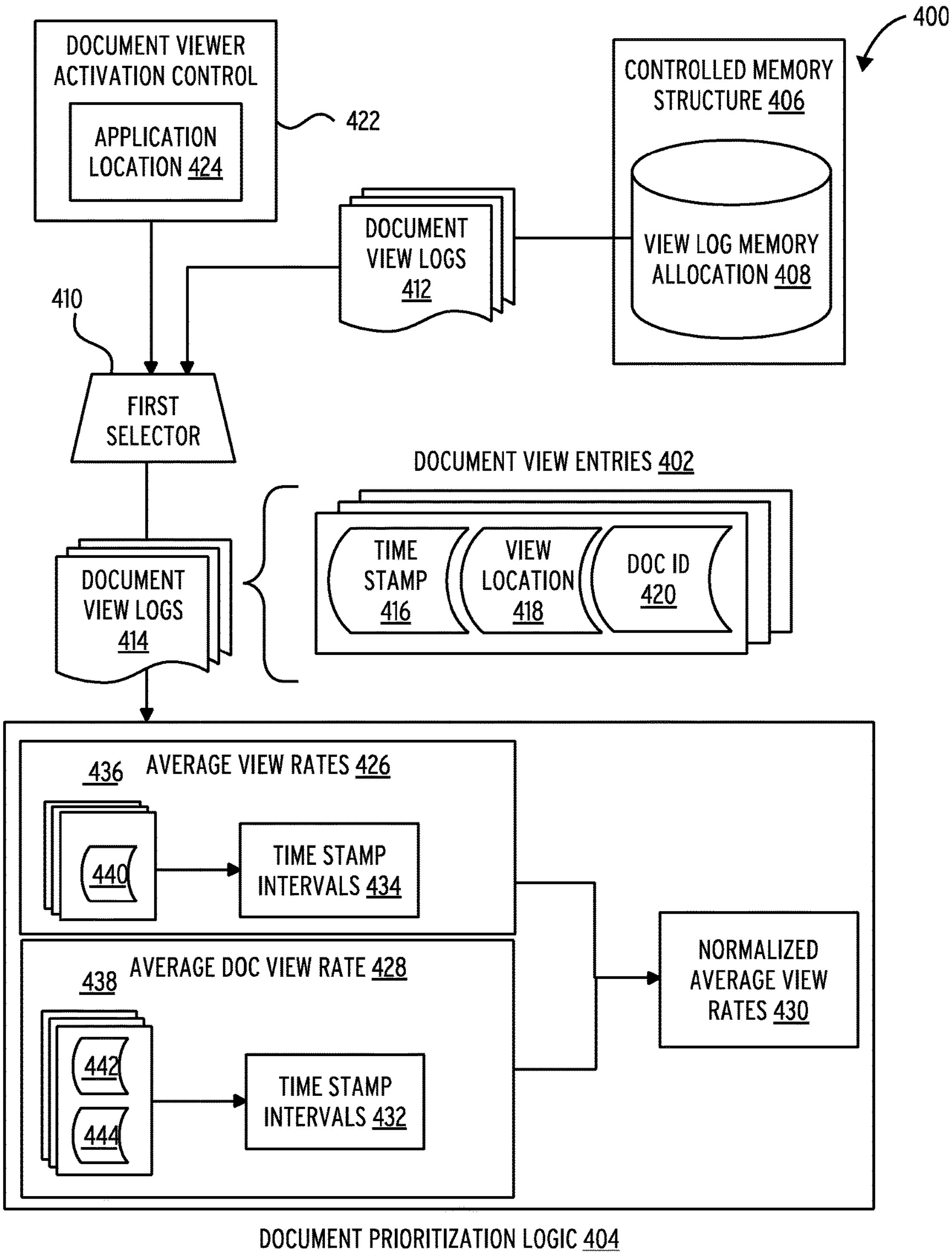

400
DOCUMENT VIEWER ACTIVATION CONTROL
APPLICATION LOCATION 424
422
CONTROLLED MEMORY STRUCTURE 406
VIEW LOG MEMORY ALLOCATION 408
DOCUMENT VIEW LOGS 412
410
FIRST SELECTOR
DOCUMENT VIEW ENTRIES 402
TIME STAMP 416
VIEW LOCATION 418
DOC ID 420
DOCUMENT VIEW LOGS 414
436
AVERAGE VIEW RATES 426
440
TIME STAMP INTERVALS 434
438
AVERAGE DOC VIEW RATE 428
442
444
TIME STAMP INTERVALS 432
NORMALIZED AVERAGE VIEW RATES 430
DOCUMENT PRIORITIZATION LOGIC 404

FIG. 4

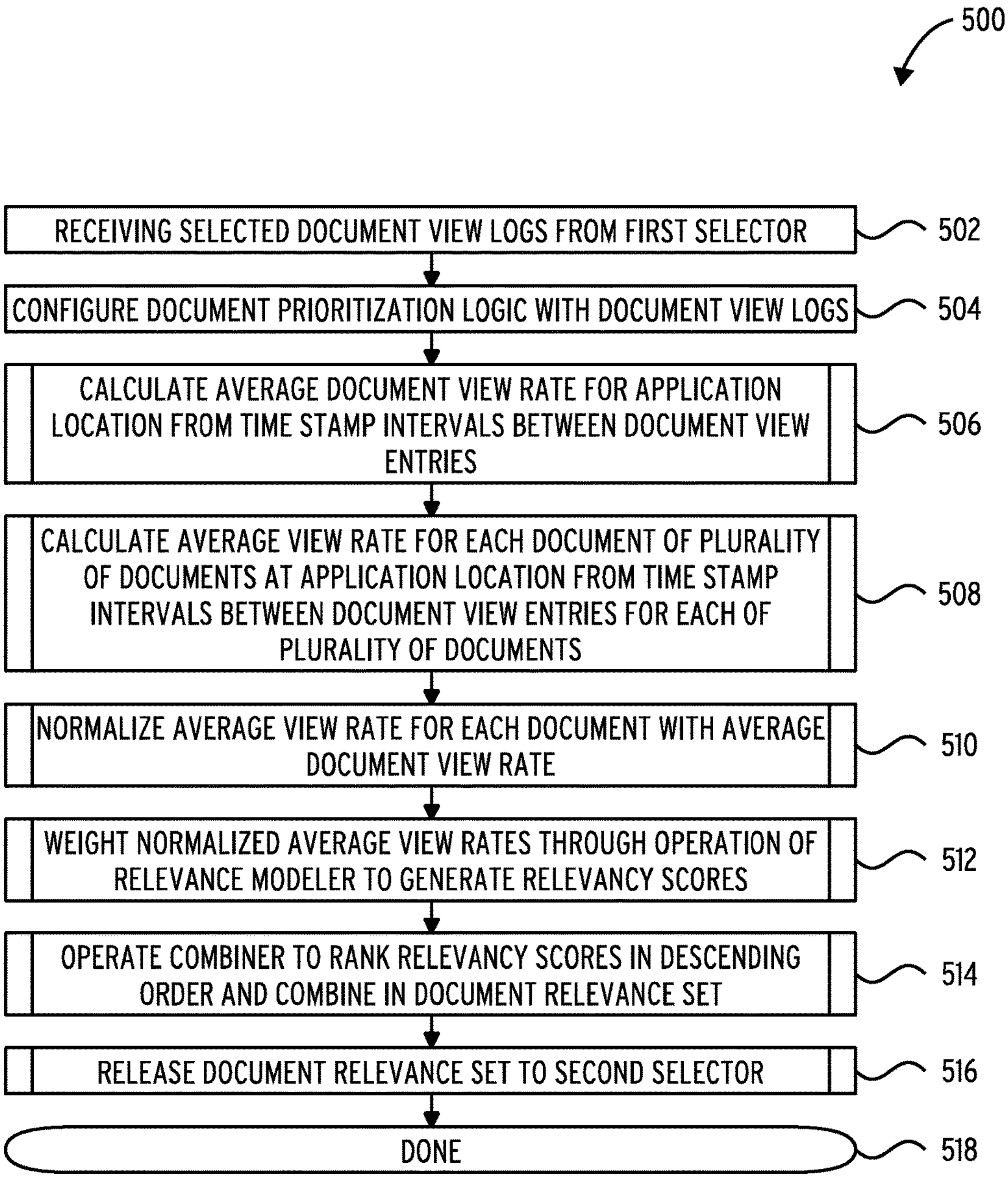

500
RECEIVING SELECTED DOCUMENT VIEW LOGS FROM FIRST SELECTOR
502
CONFIGURE DOCUMENT PRIORITIZATION LOGIC WITH DOCUMENT VIEW LOGS
504
CALCULATE AVERAGE DOCUMENT VIEW RATE FOR APPLICATION LOCATION FROM TIME STAMP INTERVALS BETWEEN DOCUMENT VIEW ENTRIES
506
CALCULATE AVERAGE VIEW RATE FOR EACH DOCUMENT OF PLURALITY OF DOCUMENTS AT APPLICATION LOCATION FROM TIME STAMP INTERVALS BETWEEN DOCUMENT VIEW ENTRIES FOR EACH OF PLURALITY OF DOCUMENTS
508
NORMALIZE AVERAGE VIEW RATE FOR EACH DOCUMENT WITH AVERAGE DOCUMENT VIEW RATE
510
WEIGHT NORMALIZED AVERAGE VIEW RATES THROUGH OPERATION OF RELEVANCE MODELER TO GENERATE RELEVANCY SCORES
512
OPERATE COMBINER TO RANK RELEVANCY SCORES IN DESCENDING ORDER AND COMBINE IN DOCUMENT RELEVANCE SET
514
RELEASE DOCUMENT RELEVANCE SET TO SECOND SELECTOR
516
DONE
518

FIG. 5

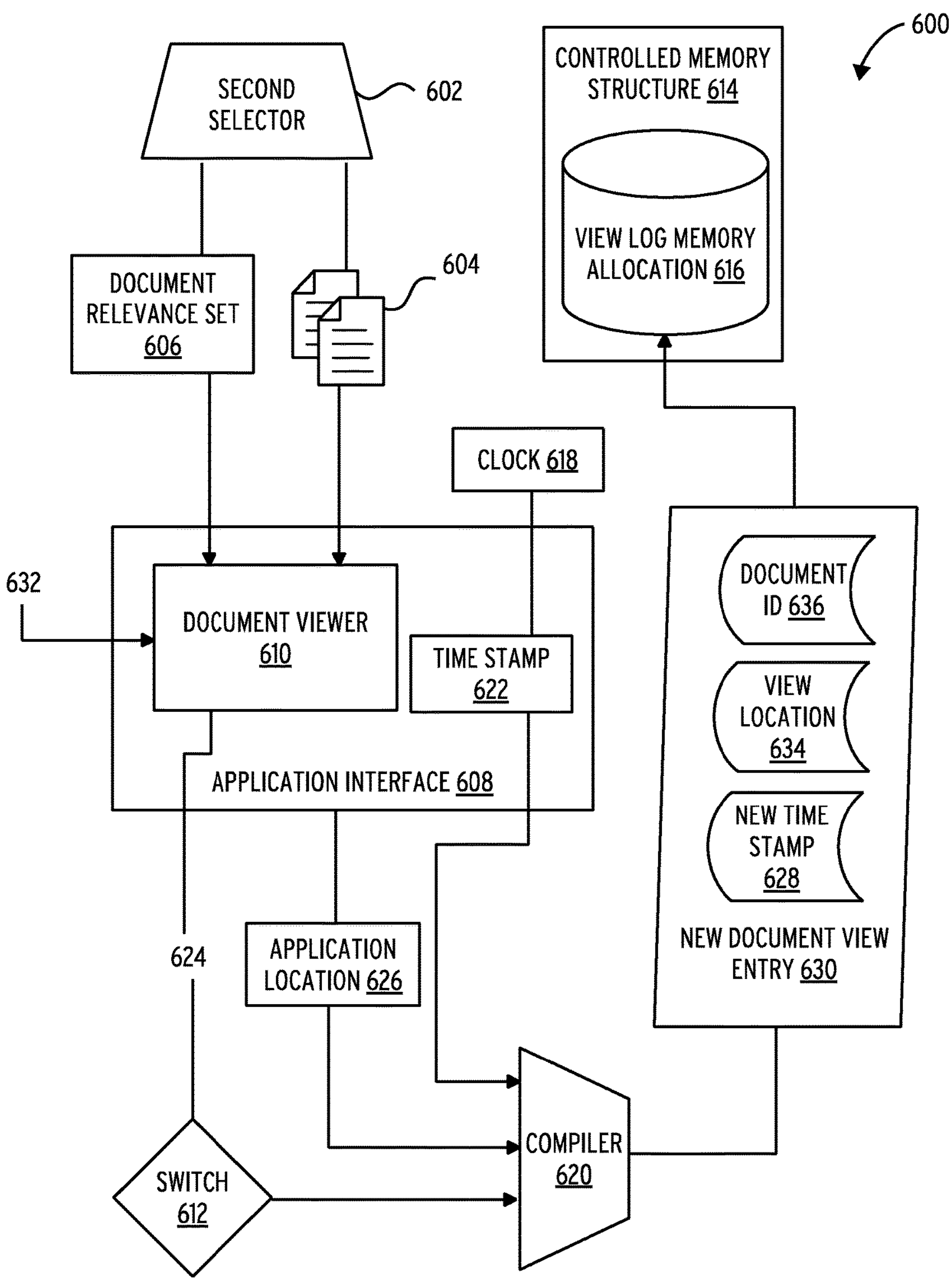

600
SECOND SELECTOR
602
CONTROLLED MEMORY STRUCTURE 614
VIEW LOG MEMORY ALLOCATION 616
DOCUMENT RELEVANCE SET 606
604
CLOCK 618
632
DOCUMENT VIEWER 610
TIME STAMP 622
APPLICATION INTERFACE 608
DOCUMENT ID 636
VIEW LOCATION 634
NEW TIME STAMP 628
NEW DOCUMENT VIEW ENTRY 630
624
APPLICATION LOCATION 626
COMPILER 620
SWITCH 612

FIG. 6

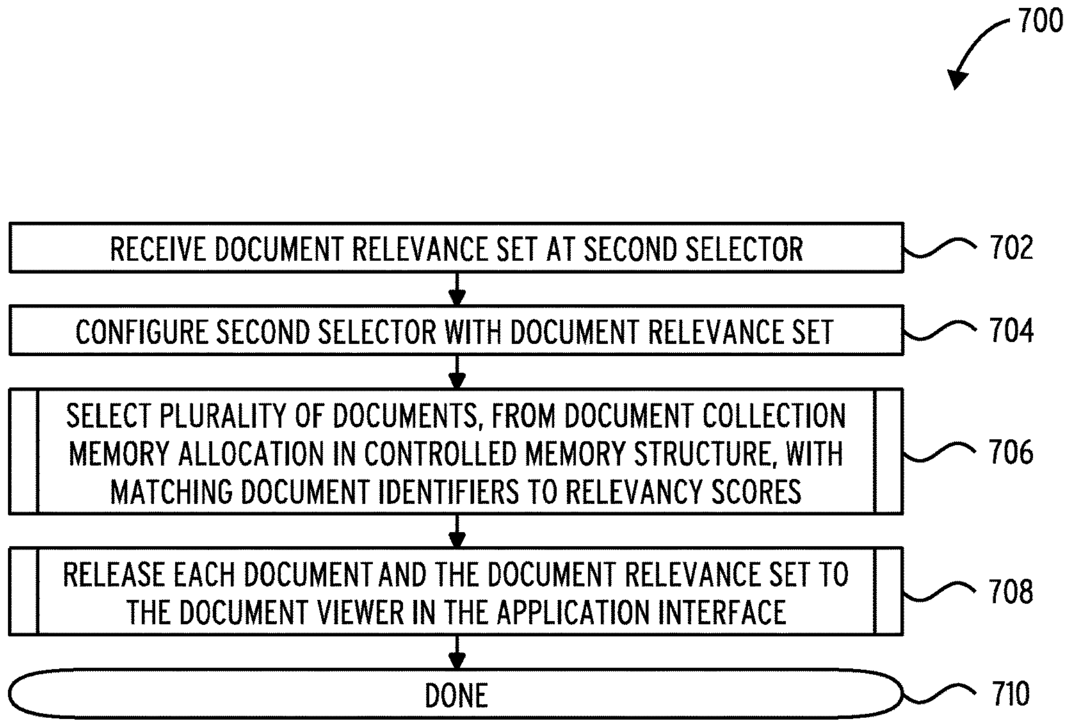

700
RECEIVE DOCUMENT RELEVANCE SET AT SECOND SELECTOR
702
CONFIGURE SECOND SELECTOR WITH DOCUMENT RELEVANCE SET
704
SELECT PLURALITY OF DOCUMENTS, FROM DOCUMENT COLLECTION MEMORY ALLOCATION IN CONTROLLED MEMORY STRUCTURE, WITH MATCHING DOCUMENT IDENTIFIERS TO RELEVANCY SCORES
706
RELEASE EACH DOCUMENT AND THE DOCUMENT RELEVANCE SET TO THE DOCUMENT VIEWER IN THE APPLICATION INTERFACE
708
DONE
710

FIG. 7

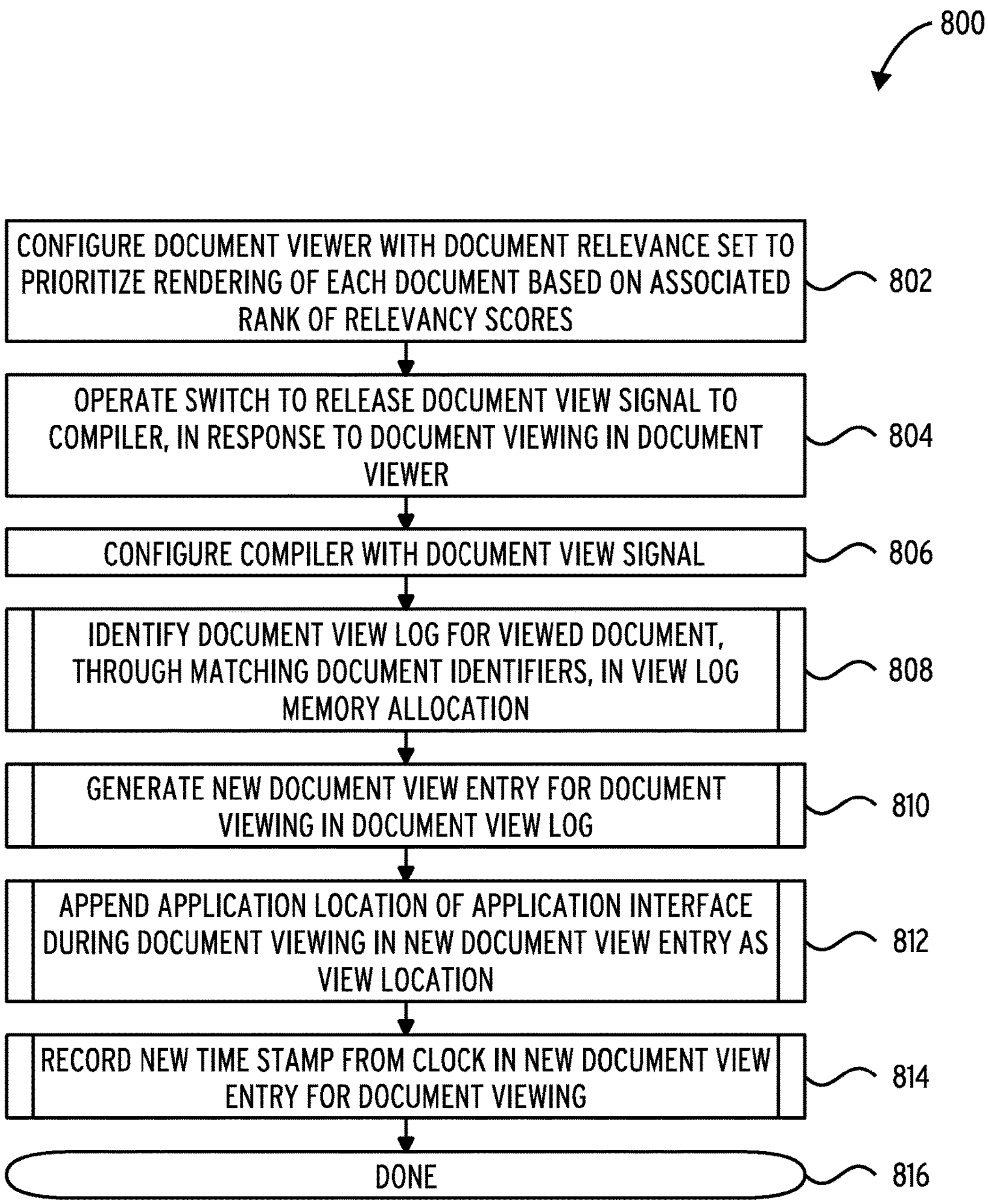

800
CONFIGURE DOCUMENT VIEWER WITH DOCUMENT RELEVANCE SET TO PRIORITIZE RENDERING OF EACH DOCUMENT BASED ON ASSOCIATED RANK OF RELEVANCY SCORES
802
OPERATE SWITCH TO RELEASE DOCUMENT VIEW SIGNAL TO COMPILER, IN RESPONSE TO DOCUMENT VIEWING IN DOCUMENT VIEWER
804
CONFIGURE COMPILER WITH DOCUMENT VIEW SIGNAL
806
IDENTIFY DOCUMENT VIEW LOG FOR VIEWED DOCUMENT, THROUGH MATCHING DOCUMENT IDENTIFIERS, IN VIEW LOG MEMORY ALLOCATION
808
GENERATE NEW DOCUMENT VIEW ENTRY FOR DOCUMENT VIEWING IN DOCUMENT VIEW LOG
810
APPEND APPLICATION LOCATION OF APPLICATION INTERFACE DURING DOCUMENT VIEWING IN NEW DOCUMENT VIEW ENTRY AS VIEW LOCATION
812
RECORD NEW TIME STAMP FROM CLOCK IN NEW DOCUMENT VIEW ENTRY FOR DOCUMENT VIEWING
814
DONE
816

FIG. 8

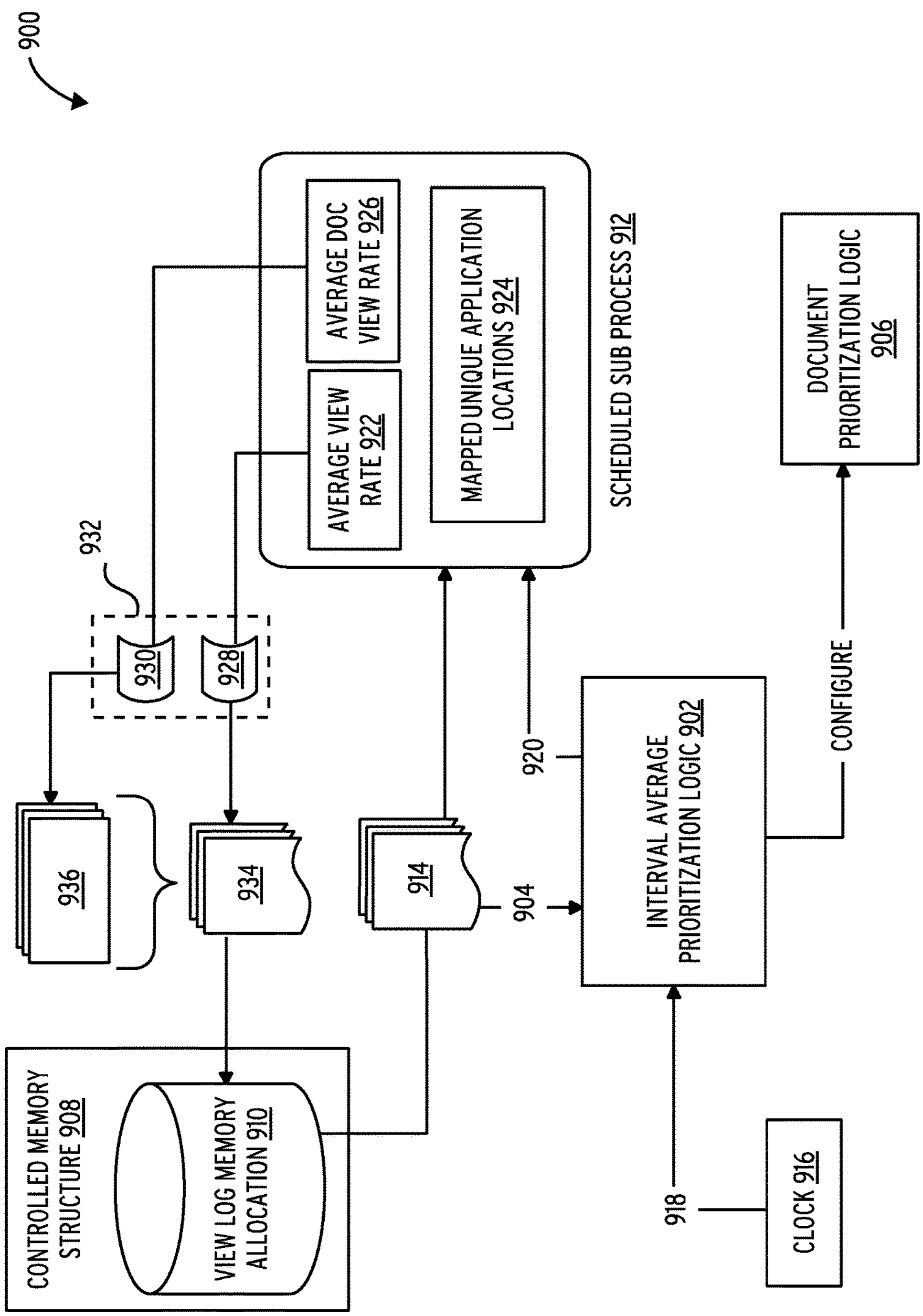

900
AVERAGE DOC VIEW RATE 926
AVERAGE VIEW RATE 922
MAPPED UNIQUE APPLICATION LOCATIONS 924
SCHEDULED SUB PROCESS 912
DOCUMENT PRIORITIZATION LOGIC 906
932
930
928
936
934
914
904
920
INTERVAL AVERAGE PRIORITIZATION LOGIC 902
CONFIGURE
CONTROLLED MEMORY STRUCTURE 908
VIEW LOG MEMORY ALLOCATION 910
918
CLOCK 916

FIG. 9

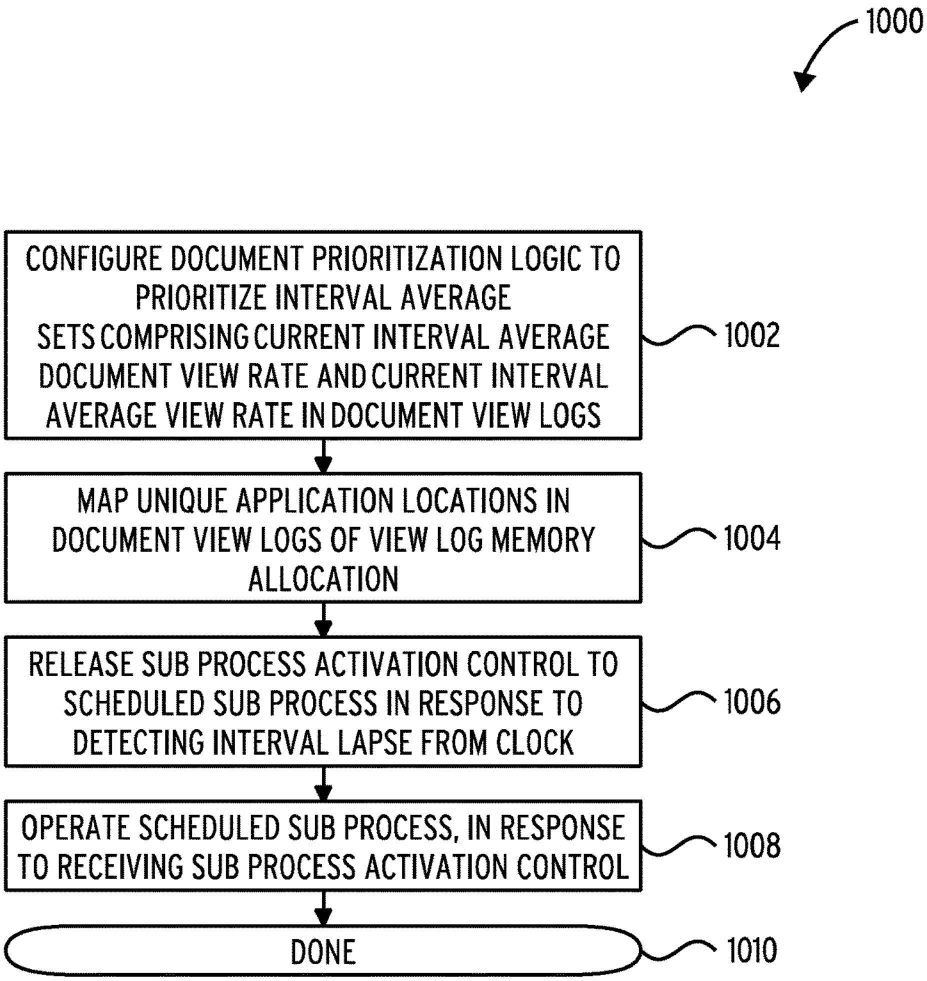

1000
CONFIGURE DOCUMENT PRIORITIZATION LOGIC TO PRIORITIZE INTERVAL AVERAGE SETS COMPRISING CURRENT INTERVAL AVERAGE DOCUMENT VIEW RATE AND CURRENT INTERVAL AVERAGE VIEW RATE IN DOCUMENT VIEW LOGS
1002
MAP UNIQUE APPLICATION LOCATIONS IN DOCUMENT VIEW LOGS OF VIEW LOG MEMORY ALLOCATION
1004
RELEASE SUB PROCESS ACTIVATION CONTROL TO SCHEDULED SUB PROCESS IN RESPONSE TO DETECTING INTERVAL LAPSE FROM CLOCK
1006
OPERATE SCHEDULED SUB PROCESS, IN RESPONSE TO RECEIVING SUB PROCESS ACTIVATION CONTROL
1008
DONE
1010

FIG. 10

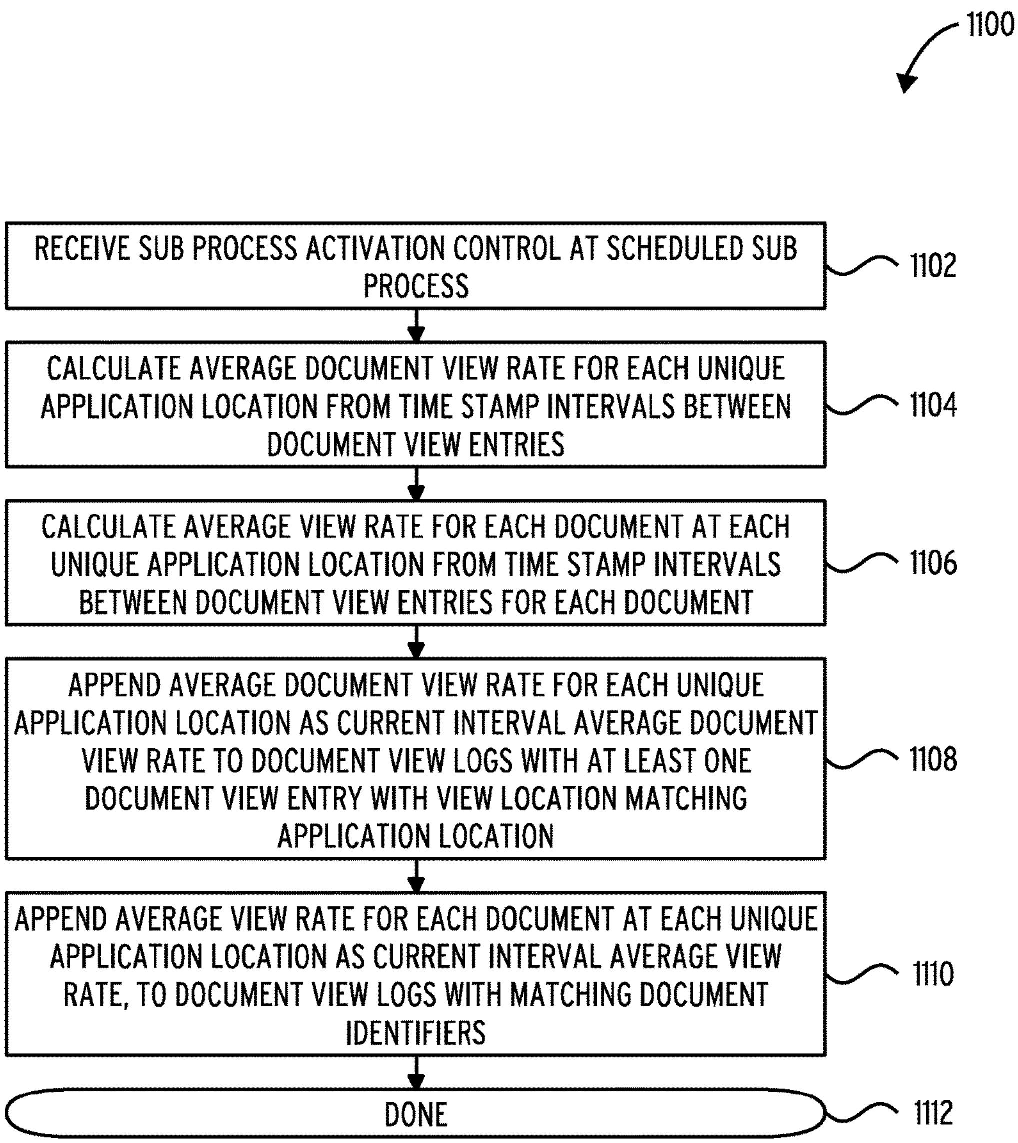

1100
RECEIVE SUB PROCESS ACTIVATION CONTROL AT SCHEDULED SUB PROCESS
1102
CALCULATE AVERAGE DOCUMENT VIEW RATE FOR EACH UNIQUE APPLICATION LOCATION FROM TIME STAMP INTERVALS BETWEEN DOCUMENT VIEW ENTRIES
1104
CALCULATE AVERAGE VIEW RATE FOR EACH DOCUMENT AT EACH UNIQUE APPLICATION LOCATION FROM TIME STAMP INTERVALS BETWEEN DOCUMENT VIEW ENTRIES FOR EACH DOCUMENT
1106
APPEND AVERAGE DOCUMENT VIEW RATE FOR EACH UNIQUE APPLICATION LOCATION AS CURRENT INTERVAL AVERAGE DOCUMENT VIEW RATE TO DOCUMENT VIEW LOGS WITH AT LEAST ONE DOCUMENT VIEW ENTRY WITH VIEW LOCATION MATCHING APPLICATION LOCATION
1108
APPEND AVERAGE VIEW RATE FOR EACH DOCUMENT AT EACH UNIQUE APPLICATION LOCATION AS CURRENT INTERVAL AVERAGE VIEW RATE, TO DOCUMENT VIEW LOGS WITH MATCHING DOCUMENT IDENTIFIERS
1110
DONE
1112

FIG. 11

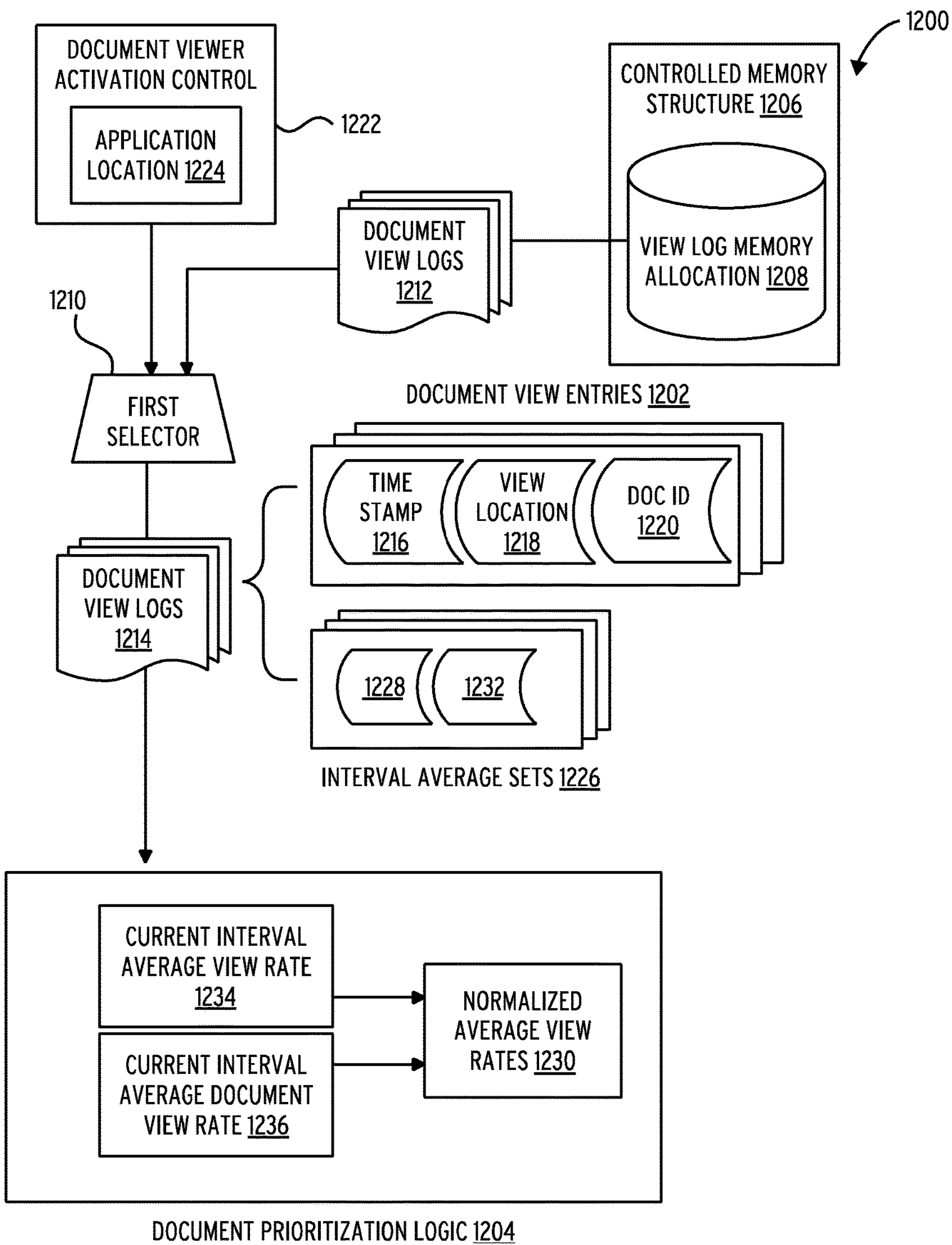

1200
DOCUMENT VIEWER ACTIVATION CONTROL
APPLICATION LOCATION 1224
1222
CONTROLLED MEMORY STRUCTURE 1206
DOCUMENT VIEW LOGS 1212
VIEW LOG MEMORY ALLOCATION 1208
1210
FIRST SELECTOR
DOCUMENT VIEW ENTRIES 1202
TIME STAMP 1216
VIEW LOCATION 1218
DOC ID 1220
DOCUMENT VIEW LOGS 1214
1228
1232
INTERVAL AVERAGE SETS 1226
CURRENT INTERVAL AVERAGE VIEW RATE 1234
CURRENT INTERVAL AVERAGE DOCUMENT VIEW RATE 1236
NORMALIZED AVERAGE VIEW RATES 1230
DOCUMENT PRIORITIZATION LOGIC 1204

FIG. 12

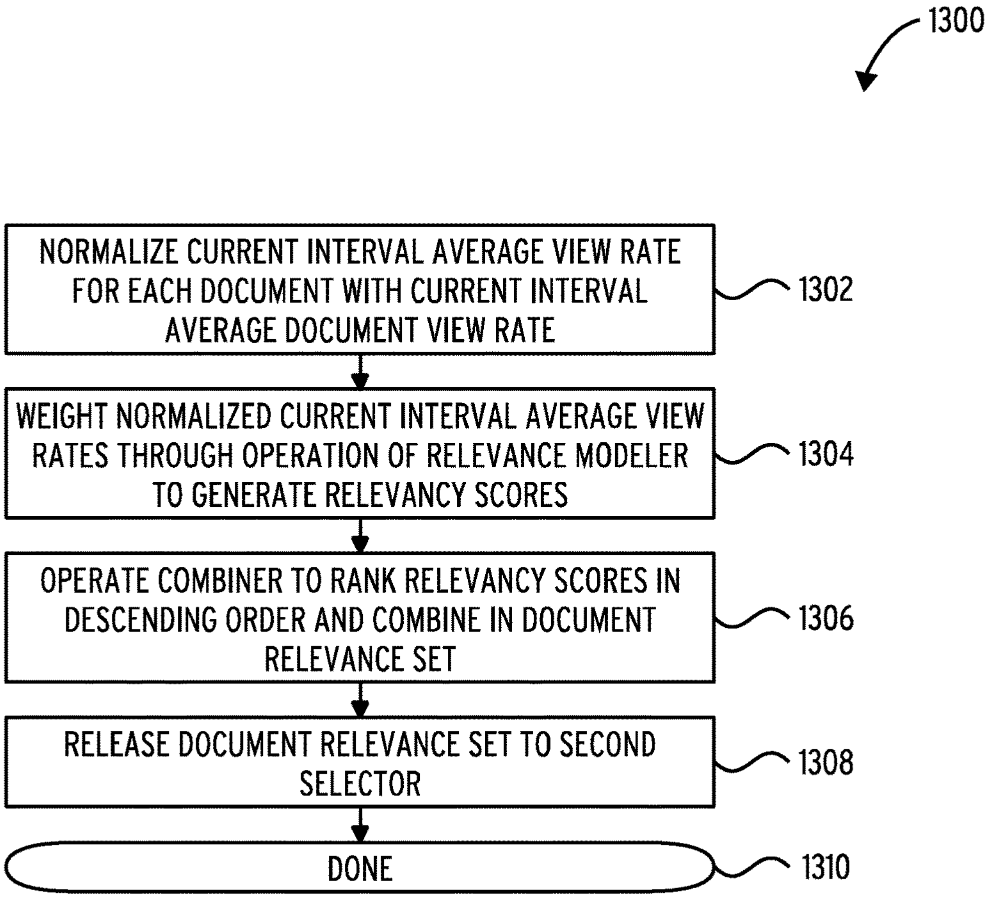

1300
NORMALIZE CURRENT INTERVAL AVERAGE VIEW RATE FOR EACH DOCUMENT WITH CURRENT INTERVAL AVERAGE DOCUMENT VIEW RATE
1302
WEIGHT NORMALIZED CURRENT INTERVAL AVERAGE VIEW RATES THROUGH OPERATION OF RELEVANCE MODELER TO GENERATE RELEVANCY SCORES
1304
OPERATE COMBINER TO RANK RELEVANCY SCORES IN DESCENDING ORDER AND COMBINE IN DOCUMENT RELEVANCE SET
1306
RELEASE DOCUMENT RELEVANCE SET TO SECOND SELECTOR
1308
DONE
1310

FIG. 13

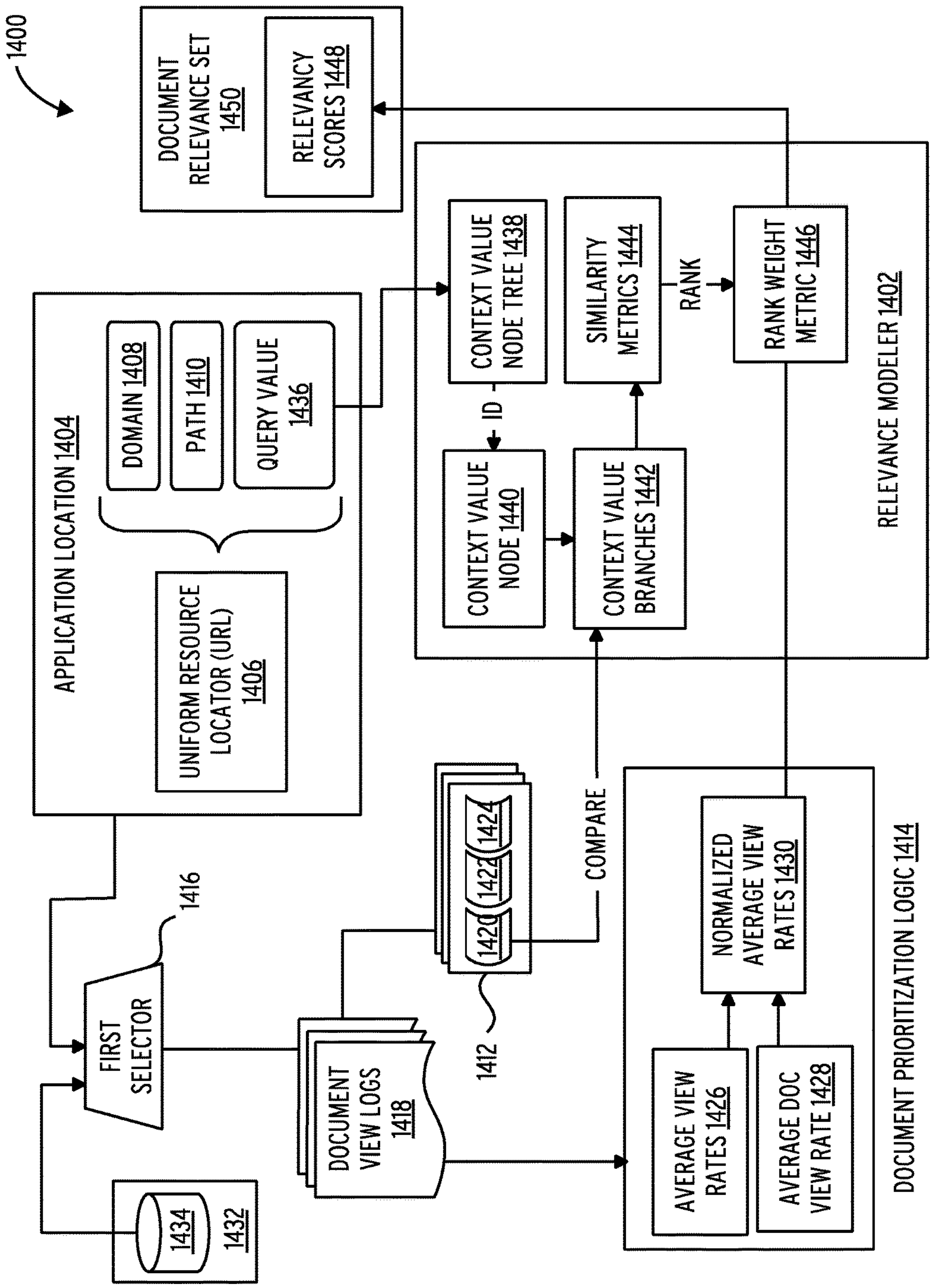

1400
DOCUMENT RELEVANCE SET 1450
RELEVANCY SCORES 1448
APPLICATION LOCATION 1404
DOMAIN 1408
PATH 1410
QUERY VALUE 1436
UNIFORM RESOURCE LOCATOR (URL) 1406
CONTEXT VALUE NODE TREE 1438
SIMILARITY METRICS 1444
RANK
RANK WEIGHT METRIC 1446
ID
CONTEXT VALUE NODE 1440
CONTEXT VALUE BRANCHES 1442
RELEVANCE MODELER 1402
1416
FIRST SELECTOR
1420
1422
1424
COMPARE
1412
DOCUMENT VIEW LOGS 1418
NORMALIZED AVERAGE VIEW RATES 1430
AVERAGE VIEW RATES 1426
AVERAGE DOC VIEW RATE 1428
DOCUMENT PRIORITIZATION LOGIC 1414
1434
1432

FIG. 14

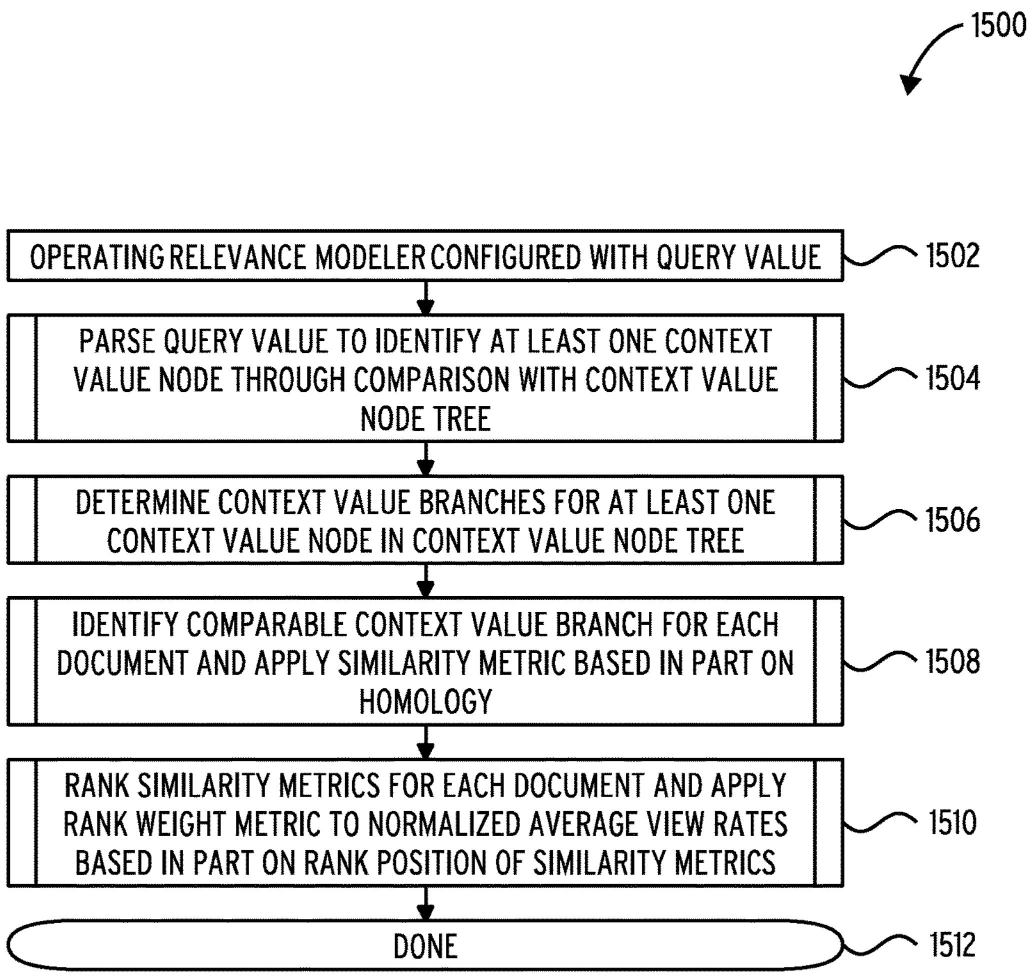

1500
OPERATING RELEVANCE MODELER CONFIGURED WITH QUERY VALUE
1502
PARSE QUERY VALUE TO IDENTIFY AT LEAST ONE CONTEXT VALUE NODE THROUGH COMPARISON WITH CONTEXT VALUE NODE TREE
1504
DETERMINE CONTEXT VALUE BRANCHES FOR AT LEAST ONE CONTEXT VALUE NODE IN CONTEXT VALUE NODE TREE
1506
IDENTIFY COMPARABLE CONTEXT VALUE BRANCH FOR EACH DOCUMENT AND APPLY SIMILARITY METRIC BASED IN PART ON HOMOLOGY
1508
RANK SIMILARITY METRICS FOR EACH DOCUMENT AND APPLY RANK WEIGHT METRIC TO NORMALIZED AVERAGE VIEW RATES BASED IN PART ON RANK POSITION OF SIMILARITY METRICS
1510
DONE
1512

FIG. 15

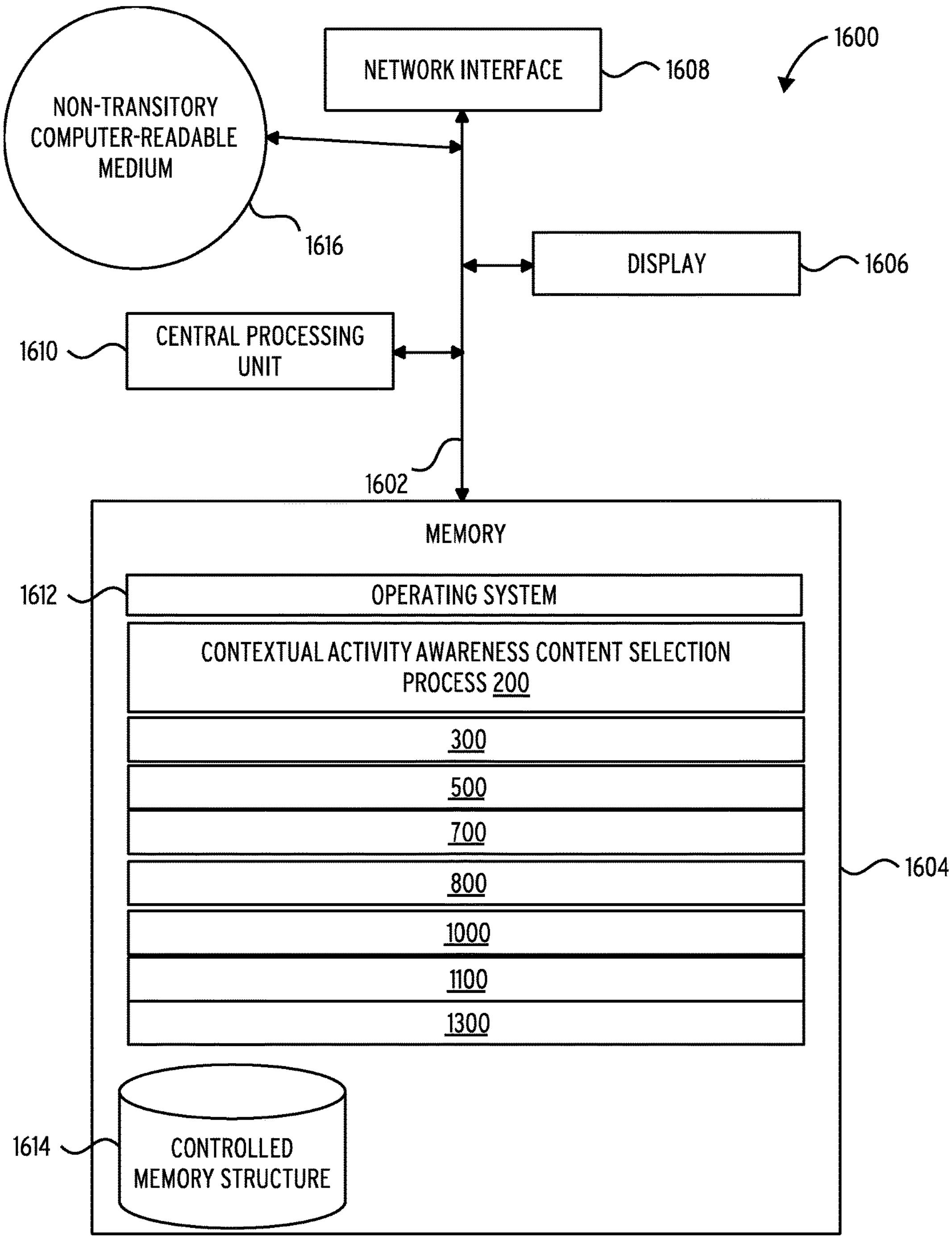

1600
NON-TRANSITORY COMPUTER-READABLE MEDIUM
1616
NETWORK INTERFACE
1608
DISPLAY
1606
1610
CENTRAL PROCESSING UNIT
1602
MEMORY
1612
OPERATING SYSTEM
CONTEXTUAL ACTIVITY AWARENESS CONTENT SELECTION PROCESS 200
300
500
700
800
1000
1100
1300
1604
1614
CONTROLLED MEMORY STRUCTURE

FIG. 16

CONTEXT- AND ACTIVITY-AWARE CONTENT SELECTION

BACKGROUND

Software applications often display content to users that might be helpful in using the application, including links to knowledge base articles, discussion forums, or help centers. Unfortunately, when there is an abundance of help content (e.g., knowledge bases can have hundreds to thousands of articles), to provide relevant content, application designers must either 1) manually select which content appears in which places in an application, 2) provide a search option to allow users to find relevant content, or 3) use some computer-based content selection method to decide which content is most relevant to display in particular situations.

In the case of automated content selection, there is a wide range of information that might be taken into account to decide what content to display. Prior art teaches of methods that segregate software applications into different screens, providing content that is specific to the screen, requiring each piece of content to be annotated with the screen to which it applies. Other prior art teaches of methods that track which content is accessed most frequently by application users, selecting only the most viewed content for display. Other methods track a specific individual user's activity, using aspects of the application that the user has engaged with to identify content that is related to those aspects.

There exists a need for methods that can combine existing prior art with the data about all of the users of an application to select the most relevant content.

BRIEF SUMMARY

In some embodiments, a machine tracking system may include a digital document viewer; logic to identify an application location corresponding to activation of the digital document viewer; a clock; a database; logic to transform the each of the group of document view log entries into an average rate of document views at the application location corresponding to activation of the digital document viewer; logic to determine a baseline rate of document view activity for the application location corresponding to activation of the digital document viewer from the average rate of document views at the application location corresponding to activation of the digital document viewer; logic to determine, for each document of a group of documents represented in the group of document view log entries, an average time interval between views of the each document of a group of documents represented in the group of document view log entries at the application location corresponding to activation of the digital document viewer; logic to normalize the average time interval between views according to the baseline rate of document view activity for the application location corresponding to activation of the digital document viewer to produce a normalized view interval; logic to weight the normalized view interval according to a sigmoidal function based on a time since the digital document rendered by the digital document viewer was last viewed to produce a relevance score for the digital document rendered by the digital document viewer; and/or logic to position a first selector for the digital document rendered by the digital document viewer in a user presentation according to the relevance score for the digital document rendered by the digital document viewer.

In some embodiments, the database may include a document view log may include a group of document view log entries.

In some embodiments, each of the group of document view log entries may include a document id corresponding to a digital document rendered by the digital document viewer, the application location corresponding to activation of the digital document viewer, and a timestamp from the clock.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

To easily identify the discussion of any particular element or act, the most significant digit or digits in a reference number refer to the figure number in which that element is first introduced.

FIG. **1** illustrates an embodiment of a contextual activity awareness content selection system **100**.

FIG. **2** illustrates an embodiment of a contextual activity awareness content selection process **200**.

FIG. **3** illustrates an embodiment of a first selector sub process **300**.

FIG. **4** illustrates an embodiment of a document prioritization logic sub system **400**.

FIG. **5** illustrates an embodiment of a document prioritization logic sub process **500**.

FIG. **6** illustrates an embodiment of a new document view entry compiler subsystem **600**.

FIG. **7** illustrates an embodiment of a second selector sub process **700**.

FIG. **8** illustrates an embodiment of a new document view entry generation sub process **800**.

FIG. **9** illustrates an embodiment of an interval average prioritization logic sub system **900**.

FIG. **10** illustrates an embodiment of an interval average prioritization logic sub process **1000**.

FIG. **11** illustrates an embodiment of a scheduled sub process **1100**.

FIG. **12** illustrates an embodiment of an interval average set prioritized document prioritization logic sub system **1200**.

FIG. **13** illustrates an embodiment of an interval average prioritized document prioritization logic sub process **1300**.

FIG. **14** illustrates an embodiment of the relevance modeler sub system **1400**.

FIG. **15** illustrates an embodiment of a relevance modeler operation sub process **1500**.

FIG. **16** illustrates an embodiment of a system **1600**.

DETAILED DESCRIPTION

Description

In some embodiments a method of contextual activity awareness content selection may include operating a switch to release a document viewer activation control to a first selector, in response to receiving an activation input for a document viewer through an application interface; configuring the first selector with the document viewer activation control; configuring the document prioritization logic with the document view logs; configuring the second selector with the document relevance set; configuring the document viewer with the document relevance set to prioritize rendering of each document based on associated rank of the relevancy scores; operating the switch to release a document view signal to a compiler, in response to a document viewing in the document viewer. The configured first selector selects the application location for the application interface from the document viewer activation control; selects document view logs, each associated with a document identifier, with document view entries each including time stamp, and view location matching the application location, from a view log memory allocation of a controlled memory structure; and/or release the document view logs to document prioritization logic. The configured document prioritization logic calculates average document view rate for the application location from time stamp intervals between the document view entries; calculates an average view rate for each document of a plurality of documents at the application location from the time stamp intervals between the document view entries for each of the plurality of documents; normalizes the average view rate for each document with the average document view rate; weights normalized average view rates through operation of a relevance modeler to generate relevancy scores; operates a combiner to rank the relevancy scores in descending order and combine in a document relevance set; and/or release the document relevance set to a second selector. The configured second selector selects the plurality of documents, from a document collection memory allocation in the controlled memory structure, with matching document identifiers to the relevancy scores; and/or releases each document and the document relevance set to the document viewer in the application interface. The configured compiler identifies a document view log for a viewed document, through the matching document identifiers, in the view log memory allocation; generates a new document view entry for the document viewing in the document view log; appends the application location of the application interface during the document viewing in the new document view entry as the view location; and/or records a new time stamp from a clock in the new document view entry for the document viewing.

In some embodiments the method of contextual activity awareness content selection may include operating an interval average prioritization logic to: configure the document prioritization logic to prioritize interval average sets including a current interval average document view rate and a current interval average view rate in the document view logs; map unique application locations in the document view logs of the view log memory allocation; release a sub process activation control to a scheduled sub process in response to detecting an interval lapse from the clock; and/or operate the scheduled sub process, in response to receiving the sub process activation control. The scheduled sub process calculates the average document view rate for each unique application location from the time stamp intervals between the document view entries; calculates the average view rate for each document at each unique application location from the time stamp intervals between the document view entries for each document; appends the average document view rate for each unique application location as the current interval average document view rate to the document view logs with at least one document view entry with the view location matching the application location; and/or appends the average view rate for each document at each unique application location as the current interval average view rate, to the document view logs with the matching document identifiers.

In some embodiments method of contextual activity awareness content selection may include configuring the document prioritization logic with the document view logs to: normalize the current interval average view rate for each document with the current interval average document view rate; weight normalized current interval average view rates through operation of the relevance modeler to generate the relevancy scores; operate the combiner to rank the relevancy scores in descending order and combine in the document relevance set; and/or release the document relevance set to the second selector.

In some embodiments method of contextual activity awareness content selection may configure the application location is a uniform resource locator (URL) to include a domain, a path, and a query value.

In some embodiments of contextual activity awareness content selection may include configuring the relevance modeler with the query value, to: parse the query value to identify at least one context value node through comparison with a context value node tree; determine context value branches for the at least one context value node in the context value node tree; identify a comparable context value branch for each document and apply a similarity metric based in part on homology; and/or rank similarity metrics for each document and apply rank weight metric to the normalized average view rates based in part on rank position of the similarity metrics.

In some embodiments, an application may provide a control such as a floating help button that a user may invoke to request relevant content. Many alternative embodiments for providing access to documents in an application are possible, including interfaces that allow users to search for documents, browsing views of all documents, tools for communicating with other users online to find helpful content, or explicit links to documents in the application itself. The present method does not depend on the method by which documents are retrieved, but that viewing the document is a discrete, observable event.

In some embodiments, the application locations may be defined in a variety of ways. In an embodiment, location may be determined by a web page's uniform resource locator (URL) or by a unique identifier associated with an application screen (as is often assigned when building an application's user interface). Other embodiments define location according to physical geolocation, the visual structure of the content on an application screen, or the textual content on the application screen. The present method does not depend on how location is defined, but that locations are discrete and may be identified unambiguously.

In some embodiments, each time a document is viewed in an application location, a document view event is logged according to timestamp and application location. A user might click on a document title to request the full document or select a link to the document. When this event occurs, via whatever means, the document view is logged.

When an application user requests relevant documents at a particular application location, the system uses the log of document views, and the user's current application location to determine which documents are likely to be most relevant to the user. First, the system determines average time between document views at the application location to establish a baseline rate of document view activity. Then, for each document in the collection, the system determines the average time interval between views of the document at the application location, normalizing the average view interval according to the baseline rate determined above. Then, the normalized view interval is weighted according to a sigmoidal function based on the time since when the document was last viewed. These calculations result in a relevance score for each document based on how frequently and recently it was viewed at the application location. Documents that are viewed recently and/or frequently get the highest scores. In the an embodiment, documents that were not viewed at the current application location are not included in scoring. In alternative embodiments, these non-viewed documents might be included, e.g. with lower weighting on the score.

In some embodiments using the relevance scores, the system sorts the documents in descending score order, resulting in a list of the most viewed questions for the application location. In the an embodiment, the documents are displayed in this descending order atop the application screen, showing an application user which documents other visitors have viewed at this location. In alternative embodiments, a new screen might be shown to display the sorted list of relevant documents, or the documents might be displayed on a different device.

In alternative embodiments, other context- or activity-sensitive factors may influence the scoring of relevant documents. For instance, metadata of the documents such as indicators of its helpfulness or importance could influence the weighting of relevant documents. Information about documents that an application user has previously viewed might also influence scoring, excluding documents that the user has viewed, or preferring documents that are related to documents that the user has viewed. Documents might be related by metadata about the documents topics or by explicit links between documents, either in the document content, or as metadata for linking related documents.

In some embodiments the system may include a document store, viewer, clock, locator, view log, interval timer, normalizer, sigmoidal weighting, and sorter. The viewer receives a document from the document store and in response renders the document on a machine interface for user viewing of the document. Due to configured interactions between the various system components, this initiates additional actions and processing.

In some embodiments the locator receives an event signal from the viewer and in response generates a location indication. The clock receives an event signal from the viewer and in response generates a time indication. The location and time indications are communicated to an event log. The view log also receives a doc id signal from the viewer and in response adds an entry for the doc id to the view log, including the time and location.

In some embodiments the interval timer receives view records from the view log and in response determines time intervals between the view records and applies the intervals to a baseline rate and average interval. The normalizer receives a baseline rate from the interval timer. The normalizer also receives an average interval from the interval timer and in response applies the average interval and baseline rate to determine the normal interval. This normal interval is then subjected to weighting.

In some embodiments the sigmoidal weighting module receives the normal interval from the normalizer and in response determines weights to associate with the normal interval, producing a location specific relevance for one or more documents. The sigmoidal weighting may also receive metadata from the document store and apply these to the weights for the location specific relevance score(s). The sorter receives a location specific relevance score signal from the sigmoidal weighting and in response sorts documents by location specific relevance.

DRAWINGS

FIG. **1** illustrates and embodiment of a contextual activity awareness content selection system **100**.

The contextual activity awareness content selection system **100** comprises an application interface **102**, a switch **104**, a first selector **120**, a document prioritization logic **134**, an interval average prioritization logic **170**, a relevance modeler **148**, a controlled memory structure **160**, a combiner **156**, a second selector **122**, a clock **110**, and a compiler **112**. The application interface **102** comprises a document viewer **130**, and an application location **168**. The document viewer **130** comprises an activation input **162**. The controlled memory structure **160** comprises a view log memory allocation **108** and a document collection memory allocation **118**. The view log memory allocation **108** comprises document view logs **124**. The document view logs **124** comprise document view entries **114** and a new document view entry **126**. The document view entries **114** each comprise, in part by association, a document identifier **138** a time stamp **140**, and a view location **142**. The new document view entry **126** comprises a time stamp **166** and a view location **164**. The switch **104** comprises a document viewer activation control **174**. The first selector **120** comprises selected document view logs **136**. The selected document view logs **136** comprise document view entries **106**, each comprising matching view location **146**. The document view logs **124** comprise interval average sets **150**. The interval average sets **150** comprise a current interval average view rate **176** and a current interval average document view rate **178**. The document prioritization logic **134** comprises normalized average view rates **154**. The relevance modeler **148** comprises relevancy scores **158**. The combiner **156** comprises document relevance set **128**. The document collection memory allocation **118** comprises a plurality of documents **116**. The second selector **122** comprises a document relevance set **128** and a plurality of documents **144** comprising matching document identifier **172**. The compiler **112** comprises a new time stamp **132** and a document view signal **152**.

The contextual activity awareness content selection system **100** may be implemented in accordance the processes described in FIG. **2**, FIG. **3**, FIG. **5**, FIG. **7**, FIG. **8**, FIG. **10**, FIG. **11**, and FIG. **13**.

FIG. **2** illustrates an embodiment of a contextual activity awareness content selection process **200**. In block **202**, contextual activity awareness content selection process **200** operates a switch to release a document viewer activation control to a first selector, in response to receiving an activation input for a document viewer through an application interface. In block **204**, contextual activity awareness content selection process **200** configures the first selector with the document viewer activation control to. In block **206**, contextual activity awareness content selection process **200** selects document view logs, each associated with a document identifier, with document view entries each. In block **208**, contextual activity awareness content selection process **200** configures the document prioritization logic with the document view logs to. In block **210**, contextual activity awareness content selection process **200** configures the second selector with the document relevance set to. In block **212**, contextual activity awareness content selection process **200** configures the document viewer with the document relevance set to prioritize rendering of each document based on associated rank of the relevancy scores. In block **214**, contextual activity awareness content selection process **200** operates the switch to release a document view signal to a compiler, in response to a document viewing in the document viewer. In block **216**, contextual activity awareness content selection process **200** configures the compiler with the document view signal to. In block **218**, contextual activity awareness content selection process **200** generates a new document view entry for a document viewing in a document view log. In done block **220**, contextual activity awareness content selection process **200** ends.

FIG. **3** illustrates an embodiment of a first selector sub process **300**. In block **302**, first selector sub process **300** receives enhanced document viewer activation control from a switch. In block **304**, first selector sub process **300** configures the first selector with the document viewer activation control. In block **306**, first selector sub process **300** operates the first selector. In subroutine block **308**, first selector sub process **300** selects an application location for the application interface from the document viewer activation control. In subroutine block **310**, first selector sub process **300** selects document view logs from a view log memory allocation of a controlled memory structure. In subroutine block **312**, first selector sub process **300** releases the selected document view logs to document prioritization logic. In done block **314**, first selector sub process **300** ends.

FIG. **4** illustrates an embodiment of a document prioritization logic sub system **400**. The document prioritization logic sub system **400** comprises a controlled memory structure **406**, a document viewer activation control **422**, a first selector **410**, and a document prioritization logic **404**. The controlled memory structure **406** comprises a view log memory allocation **408**. The view log memory allocation **408** comprises document view logs **412**. The document viewer activation control **422** comprises an application location **424**. The first selector **410** comprises selected document view logs **414** each comprising document view entries **402**. Each of the document view entries **402** comprise a time stamp **416**, a matching view location **418**, and a document identifier **420**. The document prioritization logic **404** comprises average view rates **426**, average document view rate **428**, and normalized average view rates **430**. The average view rates **426** comprises time stamp intervals **434** from document view entries **436**. The document view entries **436** comprise time stamps **440**. The average document view rate **428** comprises time stamp intervals **432** from document view entries **438** comprising time stamps **442** and same document identifier **444**.

The document prioritization logic sub system **400** may be implemented in accordance with the process described in FIG. **5**.

FIG. **5** illustrates an embodiment of a document prioritization logic sub process **500**. In block **502**, document prioritization logic sub process **500** receiving selected document view logs from the first selector. In block **504**, document prioritization logic sub process **500** configures the document prioritization logic with the document view logs to. In subroutine block **506**, document prioritization logic sub process **500** calculates average document view rate for the application location from time stamp intervals between the document view entries. In subroutine block **508**, document prioritization logic sub process **500** calculates an average view rate for each document of a plurality of documents at the application location from the time stamp intervals between the document view entries for each of the plurality of documents. In subroutine block **510**, document prioritization logic sub process **500** normalizes the average view rate for each document with the average document view rate. In subroutine block **512**, document prioritization logic sub process **500** weights normalized average view rates through operation of a relevance modeler to generate relevancy scores. In subroutine block **514**, document prioritization logic sub process **500** operates a combiner to rank the relevancy scores in descending order and combine in a document relevance set. In subroutine block **516**, document prioritization logic sub process **500** releases the document relevance set to a second selector. In done block **518**, document prioritization logic sub process **500** ends.

FIG. **6** illustrates an embodiment of a new document view entry compiler subsystem **600**. The new document view entry compiler subsystem **600** comprises a second selector **602**, a controlled memory structure **614**, an application interface **608**, a switch **612**, a clock **618**, and a compiler **620**. The application interface **608** comprises a document viewing **632**, a document viewer **610**, an application location **626**, and a time stamp **622** from the clock **618**. The document viewer **610** comprises a document relevance set **606** and a plurality of documents **604**. The switch **612** comprises a document view signal **624**. The controlled memory structure **614** comprises a view log memory allocation **616**. The new document view entry **630** comprises a matching document identifiers **636**, a view location **634**, and a new time stamp **628**.

The new document view entry compiler subsystem **600** may be operated in accordance with the process described in FIG. **7** and FIG. **8**.

FIG. **7** illustrates an embodiment of a second selector sub process **700**. In block **702**, second selector sub process **700** receives the document relevance set at the second selector. In block **704**, second selector sub process **700** configures the second selector with the document relevance set to. In subroutine block **706**, second selector sub process **700** selects the plurality of documents, from a document collection memory allocation in the controlled memory structure, with matching document identifiers to the relevancy scores. In subroutine block **708**, second selector sub process **700** releases each document and the document relevance set to the document viewer in the application interface. In done block **710**, second selector sub process **700** ends.

FIG. **8** illustrates an embodiment of a new document view entry generation sub process **800**. In block **802**, new document view entry generation sub process **800** configures the document viewer with the document relevance set to prioritize rendering of each document based on associated rank of the relevancy scores. In block **804**, new document view entry generation sub process **800** operates the switch to release a document view signal to a compiler, in response to a document viewing in the document viewer. In block **806**, new document view entry generation sub process **800** configures the compiler with the document view signal to. In subroutine block **808**, new document view entry generation sub process **800** identifies a document view log for a viewed document, through the matching document identifiers, in the view log memory allocation. In subroutine block **810**, new document view entry generation sub process **800** generates a new document view entry for the document viewing in the document view log. In subroutine block **812**, new document view entry generation sub process **800** appends the application location of the application interface during the document viewing in the new document view entry as the view location. In subroutine block **814**, new document view entry generation sub process **800** records a new time stamp from a clock in the new document view entry for the document viewing. In done block **816**, new document view entry generation sub process **800** ends.

FIG. **9** illustrates an embodiment of an interval average prioritization logic sub system **900**. The interval average prioritization logic sub system **900** comprises a scheduled sub process **912**, a controlled memory structure **908**, an interval average prioritization logic **902**, a document prioritization logic **906**, and a clock **916**. The controlled memory structure **908** comprises a view log memory allocation **910**. The view log memory allocation **910** comprises document view logs **914**. The document view logs **914** comprise unique application locations **904**. The clock **916** comprises interval lapse **918**. The interval average prioritization logic **902** comprises a sub process activation control **920**. The scheduled sub process **912** comprises mapped unique application locations **924** of the document view logs **914**. The scheduled sub process **912** comprises average view rate **922** and an average document view rate **926**. The view log memory allocation **910** comprises a document view logs **934**. The document view logs **934** comprise document view entries **936**. The document view logs **934** and the document view entries **936** comprise an interval average sets **932**. The interval average sets **932** comprises a current interval average document view rate **930** and a current interval average view rate **928**.

The interval average prioritization logic sub system **900** may be implemented in accordance with the process described in FIG. **10** and FIG. **11**.

FIG. **10** illustrates an embodiment of interval average prioritization logic sub process **1000**. In block **1002**, interval average prioritization logic sub process **1000** configures the document prioritization logic to prioritize interval average sets comprising a current interval average document view rate and a current interval average view rate in the document view logs. In block **1004**, interval average prioritization logic sub process **1000** maps unique application locations in the document view logs of the view log memory allocation. In block **1006**, interval average prioritization logic sub process **1000** releases a sub process activation control to a scheduled sub process in response to detecting an interval lapse from the clock. In block **1008**, interval average prioritization logic sub process **1000** operates the scheduled sub process, in response to receiving the sub process activation control. In done block **1010**, interval average prioritization logic sub process **1000** ends.

FIG. **11** illustrates an embodiment of a scheduled sub process **1100**. In block **1102**, scheduled sub process **1100** receives the sub process activation control at scheduled sub process. In block **1104**, scheduled sub process **1100** calculates the average document view rate for each unique application location from the time stamp intervals between the document view entries. In block **1106**, scheduled sub process **1100** calculates the average view rate for each document at each unique application location from the time stamp intervals between the document view entries for each document. In block **1108**, scheduled sub process **1100** appends the average document view rate for each unique application location as the current interval average document view rate to the document view logs with at least one document view entry with the view location matching the application location. In block **1110**, scheduled sub process **1100** appends the average view rate for each document at each unique application location as the current interval average view rate, to the document view logs with the matching document identifiers. In done block **1112**, scheduled sub process **1100** ends.

FIG. **12** illustrates an embodiment of an interval average set prioritized document prioritization logic sub system **1200**. The interval average set prioritized document prioritization logic sub system **1200** comprises a controlled memory structure **1206**, a document viewer activation control **1222**, a first selector **1210**, and a document prioritization logic **1204**. The controlled memory structure **1206** comprises a view log memory allocation **1208**. The view log memory allocation **1208** comprises document view logs **1212**. The document viewer activation control **1222** comprises an application location **1224**. The first selector **1210** comprises selected document view logs **1214**. The selected document view logs **1214** comprise document view entries **1202** and interval average sets **1226**. The interval average sets **1226** comprise current interval average document view rate **1228** and current interval average view rate **1232**. Each of the document view entries **1202** comprise a time stamp **1216**, a matching view location **1218**, and a document identifier **1220**. The document prioritization logic **1204** comprises a current interval average view rate **1234** for the application location, a current interval average document view rate **1236** for each document, and a normalized average view rates **1230**.

The interval average set prioritized document prioritization logic sub system **1200** may be implemented in accordance with the process described in FIG. **13**.

FIG. **13** illustrates an embodiment of an interval average prioritized document prioritization logic sub process **1300**. In block **1302**, interval average prioritized document prioritization logic sub process **1300** normalizes the current interval average view rate for each document with the current interval average document view rate. In block **1304**, interval average prioritized document prioritization logic sub process **1300** weights normalized current interval average view rates through operation of the relevance modeler to generate the relevancy scores. In block **1306**, interval average prioritized document prioritization logic sub process **1300** operates the combiner to rank the relevancy scores in descending order and combine in the document relevance set. In block **1308**, interval average prioritized document prioritization logic sub process **1300** releases the document relevance set to the second selector. In done block **1310**, interval average prioritized document prioritization logic sub process **1300** ends.

FIG. **14** illustrates an embodiment of a relevance modeler sub system **1400**. The **400** comprises a document relevance set **1450**, an application location **1404**, a relevance modeler **1402**, a controlled memory structure **1432**, a first selector **1416**, and a document prioritization logic **1414**. The application location **1404** comprises a uniform resource locator (URL) **1406** comprising a domain **1408**, a path **1410**, and a query value **1436**. The controlled memory structure **1432** comprises a view log memory allocation **1434**. The first selector **1416** comprises a selected document view logs **1418**. The selected document view logs **1418** comprise document view entries **1412** each comprising document identifier **1420**, a time stamp **1422**, and a view location **1424**. The document prioritization logic **1414** comprises average view rates **1426**, average document view rate **1428**, and normalized average view rates **1430**. The relevance modeler **1402** comprises a context value node **1440**, a context value node tree **1438**, a context value branches **1442**, a similarity metrics **1444**, a rank weight metric **1446**. The document relevance set **1450** comprises a relevancy scores **1448**.

The relevance modeler sub system **1400** may be implemented in accordance with the process described in FIG. **15**.

FIG. **15** illustrates an embodiment of a relevance modeler operation sub process **1500**. In block **1502**, relevance modeler operation sub process **1500** operating a relevance modeler configured with the query value. In subroutine block **1504**, relevance modeler operation sub process **1500** parses the query value to identify at least one context value node through comparison with a context value node tree. In subroutine block **1506**, relevance modeler operation sub process **1500** determines context value branches for the at least one context value node in the context value node tree. In subroutine block **1508**, relevance modeler operation sub process **1500** identifies a comparable context value branch for each document and apply a similarity metric based in part on homology. In subroutine block **1510**, relevance modeler operation sub process **1500** ranks similarity metrics for each document and apply rank weight metric to the normalized average view rates based in part on rank position of the similarity metrics. In done block **1512**, relevance modeler operation sub process **1500** ends.

FIG. **16** illustrates several components of an exemplary system **1600** in accordance with one embodiment. In various embodiments, system **1600** may include a desktop PC, server, workstation, mobile phone, laptop, tablet, set-top box, appliance, or other computing device that is capable of performing operations such as those described herein. In some embodiments, system **1600** may include many more components than those shown in FIG. **16**. However, it is not necessary that all of these generally conventional components be shown in order to disclose an illustrative embodiment. Collectively, the various tangible components or a subset of the tangible components may be referred to herein as "logic" configured or adapted in a particular way, for example as logic configured or adapted with particular software or firmware.

In various embodiments, system **1600** may comprise one or more physical and/or logical devices that collectively provide the functionalities described herein. In some embodiments, system **1600** may comprise one or more replicated and/or distributed physical or logical devices.

In some embodiments, system **1600** may comprise one or more computing resources provisioned from a "cloud computing" provider, for example, Amazon Elastic Compute Cloud ("Amazon EC2"), provided by Amazon.com, Inc. of Seattle, Wash.; Sun Cloud Compute Utility, provided by Sun Microsystems, Inc. of Santa Clara, Calif.; Windows Azure, provided by Microsoft Corporation of Redmond, Wash., and the like.

System **1600** includes a bus **1602** interconnecting several components including a network interface **1608**, a display **1606**, a central processing unit **1610**, and a memory **1604**.

Memory **1604** generally comprises a random access memory ("RAM") and permanent non-transitory mass storage device, such as a hard disk drive or solid-state drive. Memory **1604** stores an operating system **1612**, a contextual activity awareness content selection process **200**, a first selector sub process **300**, a document prioritization logic sub process **500**, a second selector sub process **700**, a new document view entry generation sub process **800**, an interval average prioritization logic sub process **1000**, a scheduled sub process **1100**, and an interval average prioritized document prioritization logic sub process **1300**.

These and other software components may be loaded into memory **1604** of system **1600** using a drive mechanism (not shown) associated with a non-transitory computer-readable medium **1616**, such as a floppy disc, tape, DVD/CD-ROM drive, memory card, or the like.

Memory **1604** also includes controlled memory structure **1614**. In some embodiments, system **1600** may communicate with controlled memory structure **1614** via network interface **1608**, a storage area network ("SAN"), a high-speed serial bus, and/or via the other suitable communication technology.

In some embodiments, controlled memory structure **1614** may comprise one or more storage resources provisioned from a "cloud storage" provider, for example, Amazon Simple Storage Service ("Amazon S3"), provided by Amazon.com, Inc. of Seattle, Wash., Google Cloud Storage, provided by Google, Inc. of Mountain View, Calif., and the like.

References to "one embodiment" or "an embodiment" do not necessarily refer to the same embodiment, although they may. Unless the context clearly requires otherwise, throughout the description and the claims, the words "comprise," "comprising," and the like are to be construed in an inclusive sense as opposed to an exclusive or exhaustive sense; that is to say, in the sense of "including, but not limited to." Words using the singular or plural number also include the plural or singular number respectively, unless expressly limited to a single one or multiple ones. Additionally, the words "herein," "above," "below" and words of similar import, when used in this application, refer to this application as a whole and not to any particular portions of this application. When the claims use the word "or" in reference to a list of two or more items, that word covers all of the following interpretations of the word: any of the items in the list, all of the items in the list and any combination of the items in the list, unless expressly limited to one or the other. "Logic" refers to machine memory circuits, non transitory machine readable media, and/or circuitry which by way of its material and/or material-energy configuration comprises control and/or procedural signals, and/or settings and values (such as resistance, impedance, capacitance, inductance, current/voltage ratings, etc.), that may be applied to influence the operation of a device. Magnetic media, electronic circuits, electrical and optical memory (both volatile and nonvolatile), and firmware are examples of logic. Logic specifically excludes pure signals or software per se (however does not exclude machine memories comprising software and thereby forming configurations of matter). Those skilled in the art will appreciate that logic may be distributed throughout one or more devices, and/or may be comprised of combinations memory, media, processing circuits and controllers, other circuits, and so on. Therefore, in the interest of clarity and correctness logic may not always be distinctly illustrated in drawings of devices and systems, although it is inherently present therein. The techniques and procedures described herein may be implemented via logic distributed in one or more computing devices. The particular distribution and choice of logic will vary according to implementation. Those having skill in the art will appreciate that there are various logic implementations by which processes and/or systems described herein can be effected (e.g., hardware, software, and/or firmware), and that the preferred vehicle will vary with the context in which the processes are deployed. "Software" refers to logic that may be readily readapted to different purposes (e.g. read/write volatile or nonvolatile memory or media). "Firmware" refers to logic embodied as read-only memories and/or media. Hardware refers to logic embodied as analog and/or digital circuits. If an implementer determines that speed and accuracy are paramount, the implementer may opt for a hardware and/or firmware vehicle; alternatively, if flexibility is paramount, the implementer may opt for a solely software implementation; or, yet again alternatively, the implementer may opt for some combination of hardware, software, and/or firmware. Hence, there are several possible vehicles by which the processes described herein may be effected, none of which is inherently superior to the other in that any vehicle to be utilized is a choice dependent upon the context in which the vehicle will be deployed and the specific concerns (e.g., speed, flexibility, or predictability) of the implementer, any of which may vary. Those skilled in the art will recognize that optical aspects of implementations may involve optically-oriented hardware, software, and or firmware. The foregoing detailed description has set forth various embodiments of the devices and/or processes via the use of block diagrams, flowcharts, and/or examples. Insofar as such block diagrams, flowcharts, and/or examples contain one or more functions and/or operations, it will be understood as notorious by those within the art that each function and/or operation within such block diagrams, flowcharts, or examples can be implemented, individually and/or collectively, by a wide range of hardware, software, firmware, or virtually any combination thereof. Several portions of the subject matter described herein may be implemented via Application Specific Integrated Circuits (ASICs), Field Programmable Gate Arrays (FPGAs), digital signal processors (DSPs), or other integrated formats. However, those skilled in the art will recognize that some aspects of the embodiments disclosed herein, in whole or in part, can be equivalently implemented in standard integrated circuits, as one or more computer programs running on one or more computers (e.g., as one or more programs running on one or more computer systems), as one or more programs running on one or more processors (e.g., as one or more programs running on one or more microprocessors), as firmware, or as virtually any combination thereof, and that designing the circuitry and/or writing the code for the software and/or firmware would be well within the skill of one of skill in the art in light of this disclosure. In addition, those skilled in the art will appreciate that the mechanisms of the subject matter described herein are capable of being distributed as a program product in a variety of forms, and that an illustrative embodiment of the subject matter described herein applies equally regardless of the particular type of signal bearing media used to actually carry out the distribution. Examples of a signal bearing media include, but are not limited to, the following: recordable type media such as floppy disks, hard disk drives, CD ROMs, digital tape, flash drives, SD cards, solid state fixed or removable storage, and computer memory. In a general sense, those skilled in the art will recognize that the various aspects described herein which can be implemented, individually and/or collectively, by a wide range of hardware, software, firmware, or any combination thereof can be viewed as being composed of various types of "circuitry." Consequently, as used herein "circuitry" includes, but is not limited to, electrical circuitry having at least one discrete electrical circuit, electrical circuitry having at least one integrated circuit, electrical circuitry having at least one application specific integrated circuit, circuitry forming a general purpose computing device configured by a computer program (e.g., a general purpose computer configured by a computer program which at least partially carries out processes and/or devices described herein, or a microprocessor configured by a computer program which at least partially carries out processes and/or devices described herein), circuitry forming a memory device (e.g., forms of random access memory), and/or circuitry forming a communications device (e.g., a modem, communications switch, or optical-electrical equipment). Those skilled in the art will recognize that it is common within the art to describe devices and/or processes in the fashion set forth herein, and thereafter use standard engineering practices to integrate such described devices and/or processes into larger systems. That is, at least a portion of the devices and/or processes described herein can

be integrated into a network processing system via a reasonable amount of experimentation.

What is claimed is:

1. A method comprising:

operating a switch to release a document viewer activation control to a first selector, in response to receiving an activation input for a document viewer through an application interface;

configuring the first selector with the document viewer activation control to:

- select application location for the application interface from the document viewer activation control;
- select document view logs, each associated with a document identifier, with document view entries each comprising a time stamp, and view location matching the application location, from a view log memory allocation of a controlled memory structure; and
- release the document view logs to document prioritization logic;

configuring the document prioritization logic with the document view logs to:

- calculate average document view rate for the application location from time stamp intervals between the document view entries;
- calculate an average view rate for each document of a plurality of documents at the application location from the time stamp intervals between the document view entries for each of the plurality of documents;
- normalize the average view rate for each document with the average document view rate;
- weight normalized average view rates through operation of a relevance modeler to generate relevancy scores;
- operate a combiner to rank the relevancy scores in descending order and combine in a document relevance set; and
- release the document relevance set to a second selector;

configuring the second selector with the document relevance set to:

- select the plurality of documents, from a document collection memory allocation in the controlled memory structure, with matching document identifiers to the relevancy scores; and
- release each document and the document relevance set to the document viewer in the application interface;

configuring the document viewer with the document relevance set to prioritize rendering of each document based on associated rank of the relevancy scores;

operating the switch to release a document view signal to a compiler, in response to a document viewing in the document viewer; and

configuring the compiler with the document view signal to:

- identify a document view log for a viewed document, through the matching document identifiers, in the view log memory allocation;
- generate a new document view entry for the document viewing in the document view log;
- append the application location of the application interface during the document viewing in the new document view entry as the view location; and
- record a new time stamp from a clock in the new document view entry for the document viewing.

2. The method of claim 1, comprises:

operating an interval average prioritization logic to:

configure the document prioritization logic to prioritize interval average sets comprising a current interval average document view rate and a current interval average view rate in the document view logs;

map unique application locations in the document view logs of the view log memory allocation;

release a sub process activation control to a scheduled sub process in response to detecting an interval lapse from the clock; and

operate the scheduled sub process, in response to receiving the sub process activation control, to:

calculate the average document view rate for each unique application location from the time stamp intervals between the document view entries;

calculate the average view rate for each document at each unique application location from the time stamp intervals between the document view entries for each document;

append the average document view rate for each unique application location as the current interval average document view rate to the document view logs with at least one document view entry with the view location matching the application location; and

append the average view rate for each document at each unique application location as the current interval average view rate, to the document view logs with the matching document identifiers.

3. The method of claim 2 comprises:

configuring the document prioritization logic with the document view logs to:

normalize the current interval average view rate for each document with the current interval average document view rate;

weight normalized current interval average view rates through operation of the relevance modeler to generate the relevancy scores;

operate the combiner to rank the relevancy scores in descending order and combine in the document relevance set; and

release the document relevance set to the second selector.

4. The method of claim 1, wherein the application location is a uniform resource locator (URL) comprising a domain, a path, and a query value.

5. The method of claim 4 comprises:

configuring the relevance modeler with the query value, to:

parse the query value to identify at least one context value node through comparison with a context value node tree;

determine context value branches for the at least one context value node in the context value node tree;

identify a comparable context value branch for each document and apply a similarity metric based in part on homology; and

rank similarity metrics for each document and apply rank weight metric to the normalized average view rates based in part on rank position of the similarity metrics.

6. A non-transitory computer-readable storage medium, the computer-readable storage medium including instructions that when executed by a computer, cause the computer to:

operate a switch to release a document viewer activation control to a first selector, in response to receiving an activation input for a document viewer through an application interface;

configure the first selector with the document viewer activation control to:

select application location for the application interface from the document viewer activation control;

select document view logs, each associated with a document identifier, with document view entries each comprising a time stamp, and view location matching the application location, from a view log memory allocation of a controlled memory structure; and

release the document view logs to document prioritization logic;

configure the document prioritization logic with the document view logs to:

calculate average document view rate for the application location from time stamp intervals between the document view entries;

calculate an average view rate for each document of a plurality of documents at the application location from the time stamp intervals between the document view entries for each of the plurality of documents;

normalize the average view rate for each document with the average document view rate;

weight normalized average view rates through operation of a relevance modeler to generate relevancy scores;

operate a combiner to rank the relevancy scores in descending order and combine in a document relevance set; and

release the document relevance set to a second selector;

configure the second selector with the document relevance set to:

select the plurality of documents, from a document collection memory allocation in the controlled memory structure, with matching document identifiers to the relevancy scores; and

release each document and the document relevance set to the document viewer in the application interface;

configure the document viewer with the document relevance set to prioritize rendering of each document based on associated rank of the relevancy scores;

operate the switch to release a document view signal to a compiler, in response to a document viewing in the document viewer; and

configure the compiler with the document view signal to:

identify a document view log for a viewed document, through the matching document identifiers, in the view log memory allocation;

generate a new document view entry for the document viewing in the document view log;

append the application location of the application interface during the document viewing in the new document view entry as the view location; and

record a new time stamp from a clock in the new document view entry for the document viewing.

7. The computer-readable storage medium of claim 6, comprises:

operate an interval average prioritization logic to:

configure the document prioritization logic to prioritize interval average sets comprising a current interval average document view rate and a current interval average view rate in the document view logs;

map unique application locations in the document view logs of the view log memory allocation;

release a sub process activation control to a scheduled sub process in response to detecting an interval lapse from the clock; and

operate the scheduled sub process, in response to receiving the sub process activation control, to:

calculate the average document view rate for each unique application location from the time stamp intervals between the document view entries;

calculate the average view rate for each document at each unique application location from the time stamp intervals between the document view entries for each document;

append the average document view rate for each unique application location as the current interval average document view rate to the document view logs with at least one document view entry with the view location matching the application location; and

append the average view rate for each document at each unique application location as the current interval average view rate, to the document view logs with the matching document identifiers.

8. The computer-readable storage medium of claim **7** comprises:

configure the document prioritization logic with the document view logs to:

normalize the current interval average view rate for each document with the current interval average document view rate;

weight normalized current interval average view rates through operation of the relevance modeler to generate the relevancy scores;

operate the combiner to rank the relevancy scores in descending order and combine in the document relevance set; and

release the document relevance set to the second selector.

9. The computer-readable storage medium of claim **6**, wherein the application location is a uniform resource locator (URL) comprising a domain, a path, and a query value.

10. The computer-readable storage medium of claim **9** comprises:

configure the relevance modeler with the query value, to:

parse the query value to identify at least one context value node through comparison with a context value node tree;

determine context value branches for the at least one context value node in the context value node tree;

identify a comparable context value branch for each document and apply a similarity metric based in part on homology; and

rank similarity metrics for each document and apply rank weight metric to the normalized average view rates based in part on rank position of the similarity metrics.

11. A computer system, comprises:

a processor; and

a memory storing instructions that, when executed by the processor, configure the apparatus to:

operate a switch to release a document viewer activation control to a first selector, in response to receiving an activation input for a document viewer through an application interface;

configure the first selector with the document viewer activation control to:

select application location for the application interface from the document viewer activation control;

select document view logs, each associated with a document identifier, with document view entries each comprising a time stamp, and view location matching the application location, from a view log memory allocation of a controlled memory structure; and

release the document view logs to document prioritization logic;

configure the document prioritization logic with the document view logs to:

calculate average document view rate for the application location from time stamp intervals between the document view entries;

calculate an average view rate for each document of a plurality of documents at the application location from the time stamp intervals between the document view entries for each of the plurality of documents;

normalize the average view rate for each document with the average document view rate;

weight normalized average view rates through operation of a relevance modeler to generate relevancy scores;

operate a combiner to rank the relevancy scores in descending order and combine in a document relevance set; and

release the document relevance set to a second selector;

configure the second selector with the document relevance set to:

select the plurality of documents, from a document collection memory allocation in the controlled memory structure, with matching document identifiers to the relevancy scores; and

release each document and the document relevance set to the document viewer in the application interface;

configure the document viewer with the document relevance set to prioritize rendering of each document based on associated rank of the relevancy scores;

operate the switch to release a document view signal to a compiler, in response to a document viewing in the document viewer; and

configure the compiler with the document view signal to:

identify a document view log for a viewed document, through the matching document identifiers, in the view log memory allocation;

generate a new document view entry for the document viewing in the document view log;

append the application location of the application interface during the document viewing in the new document view entry as the view location; and

record a new time stamp from a clock in the new document view entry for the document viewing.

12. The computer system as claimed in claim **11**, comprises:

operate an interval average prioritization logic to:

configure the document prioritization logic to prioritize interval average sets comprising a current interval average document view rate and a current interval average view rate in the document view logs;

map unique application locations in the document view logs of the view log memory allocation;

release a sub process activation control to a scheduled sub process in response to detecting an interval lapse from the clock; and

operate the scheduled sub process, in response to receiving the sub process activation control, to:

calculate the average document view rate for each unique application location from the time stamp intervals between the document view entries;

calculate the average view rate for each document at each unique application location from the time stamp intervals between the document view entries for each document;

append the average document view rate for each unique application location as the current interval average document view rate to the document view logs with at least one document view entry with the view location matching the application location; and

append the average view rate for each document at each unique application location as the current interval average view rate, to the document view logs with the matching document identifiers.

13. The computer system as claimed in claim **12** comprises:

configure the document prioritization logic with the document view logs to:

normalize the current interval average view rate for each document with the current interval average document view rate;

weight normalized current interval average view rates through operation of the relevance modeler to generate the relevancy scores;

operate the combiner to rank the relevancy scores in descending order and combine in the document relevance set; and

release the document relevance set to the second selector.

14. The computer system as claimed in claim **11**, wherein the application location is a uniform resource locator (URL) comprising a domain, a path, and a query value.

15. The computer system as claimed in claim **14** comprises:

configure the relevance modeler with the query value, to:

parse the query value to identify at least one context value node through comparison with a context value node tree;

determine context value branches for the at least one context value node in the context value node tree;

identify a comparable context value branch for each document and apply a similarity metric based in part on homology; and

rank similarity metrics for each document and apply rank weight metric to the normalized average view rates based in part on rank position of the similarity metrics.

* * * * *

File name: US9727561B1.docx

Art Unit Predictions

Statistics for the five most-likely results, in decreasing order

Art Unit	Allowance rate	Pendency (months)	Avg. no. of Office actions	% granted with appeal
2177	56%	49	2.7	11%
2154	81%	43	2.6	12%
2176	57%	49	2.7	11%
2144	67%	44	2.6	7%
2178	61%	51	2.7	8%

§101

Eligibility based on similarity to claims rejected under 101 for abstraction

Low ELIGIBILITY High

Words related to low eligibility:

average	*rate*	*relevancy*	*allocation*
selector	*intervals*	*for*	*prioritize*
descending	*each*	*entries*	*rank*
normalized	*stamp*	*rates*	*release*
order	*combine*		

§102 §103

Novelty based on most-similar pieces of art

Related patent documents

9727561	7171619	9448987	20150324451
20140279716	9678957	8195654	7711679
8290927	6557015		

§112

Clarity issues based on language defects in the application

0	0	0	0
Antecedent basis issues	Figure reference issues	Unsupported claim terms	Claim order and format issues

Antecedent-basis issues

RoboReview™ found no antecedent-basis issues

Claim support issues

RoboReview™ found no claim support issues

Claim number and format issues

RoboReview™ found no claim number and format issues

Parts list

RoboReview™ found no issues with numbered parts