



(51) International Patent Classification:
G06Q 50/22 (2012.01)

(21) International Application Number:
PCT/US2013/054836

(22) International Filing Date:
14 August 2013 (14.08.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/720,611 31 October 2012 (31.10.2012) US
13/790,379 8 March 2013 (08.03.2013) US

(71) Applicant: ORACLE INTERNATIONAL CORPORATION [US/US]; 500 Oracle Parkway, Redwood Shores, CA 94065 (US).

(72) Inventors: TRIEBEL, William, Andrew; 2004 Eddy Street, San Francisco, CA 94115 (US). BRAUNBOUGHOS, Michael, Lawrence; Thujaweg 13, 80939 Munich (DE).

(74) Agents: MULLERVY, Keith, M. et al.; Miles & Stockbridge P.C., 1751 Pinnacle Drive, Suite 500, Mclean, VA 22102-3833 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— with declaration under Article 17(2)(a); without abstract; title not checked by the International Searching Authority



WO 2014/070278 A2

(54) Title: INTEROPERABLE CASE SERIES SYSTEM

(57) Abstract:

INTEROPERABLE CASE SERIES SYSTEM

CROSS-REFERENCE TO RELATED APPLICATIONS:

[0001] This application claims priority of U.S. Provisional Patent Application Ser. No. 61/720,611, filed on October 31, 2012, the subject matter of which is hereby incorporated by reference.

FIELD:

[0002] One embodiment is directed to a computer system, and more particularly, to a computer system that manages clinical data.

BACKGROUND:

[0003] In the domain of health sciences, such as drug safety, a “case series” is a list of adverse event cases. An “adverse event case” or “case” is a data record of a particular incident of an adverse event occurring in a patient. Each adverse event case can have a unique identifier. In health science in general, a case series can also be identified as a “patient list,” or “subject list.”

[0004] Many drug safety systems can produce and consume case series, where a “drug safety system” is a system that stores drug safety data, and where drug safety data includes data related to the safety of one or more drugs, such as one or more adverse event cases. Typically, inside these systems, an executable process produces a case series and passes it to one or more executable processes which operate on it. In a common scenario, an executable process that executes a query on a data source, such as a drug safety system, can produce a case series and pass it to an executable process that executes a report. In this scenario, the case series

is the result of the executable process that executes the query on the data source. In other words, a list of cases that comprise the case series is a list of cases that matches the conditions specified in the query that is executed by the executable process on the data source. The report can be the desired output format of the case series, and the report can be executed by an executable process. The case series can typically contain at least the unique identifier (typically identified as a “case identifier”) for each case, and may also contain additional case data or metadata that represent the adverse event cases in the case series. The data fields in the case series do not necessarily have to be the same as the data fields specified in the query or in the report. The data fields in the case series can typically be fixed while the report data fields can be changed depending on the desired output format.

[0005] Many drug safety systems have multiple software applications or systems that can operate on the same drug safety data. Furthermore, it can be common to have multiple databases of the same cases organized for different purposes. For example, the United States Food and Drug Administration (“USFDA” or “FDA”) has a network of at least four drug safety databases, and has at least five major drug safety software applications/systems that can access the data.

SUMMARY:

[0006] One embodiment is an interoperable case series system that integrates a case series repository with one or more software applications. The interoperable case series system receives a case series, where the case series includes one or more adverse event cases, and where each adverse event case includes a data record that represents an adverse event. The interoperable case series system further receives a case revision, where the case revision includes case revision

information, and where the case revision information includes at least one change to an adverse event case of the case series. The interoperable case series system further stores the case series and the case revision within a case series repository using a case series data model, where the case series data model defines a format of the case series and the case revision within the case series repository. The interoperable case series system further associates the case revision with the case series using the case series data model. The interoperable case series system further retrieves the case series and the associated case revision using a case series programming interface, where the case series application programming interface defines a format of the case series and the associated case series revision for a software application.

BRIEF DESCRIPTION OF THE DRAWINGS:

[0007] Further embodiments, details, advantages, and modifications will become apparent from the following detailed description of the preferred embodiments, which is to be taken in conjunction with the accompanying drawings.

[0008] Fig. 1 illustrates a block diagram of a system that can implement an embodiment of the invention.

[0009] Fig. 2 illustrates a block diagram of an interoperable case series system, according to an embodiment of the invention.

[0010] Fig. 3 illustrates a block diagram of a case series data model, according to an embodiment of the invention.

[0011] Fig. 4 illustrates a block diagram of an example implementation of an interoperable case series system, according to an embodiment of the invention.

[0012] Fig. 5 illustrates a flow diagram of the functionality of an interoperable

case series module, according to an embodiment of the invention.

[0013] Fig. 6 illustrates a block diagram of a cohort identification system, according to an embodiment of the invention.

[0014] Fig. 7 illustrates a block diagram of an example implementation of a cohort identification system, according to an embodiment of the invention.

[0015] Fig. 8 illustrates a flow diagram of the functionality of a cohort identification module, according to an embodiment of the invention.

[0016] Fig. 9 illustrates an example query that creates an example case series that creates an example report, according to an embodiment of the invention.

DETAILED DESCRIPTION:

[0017] In one embodiment, an interoperable case series system is provided that can integrate a case series repository with one or more software applications that produce and consume case series. As described in this specification, a “computer application,” “software application,” or “application” is any collection of computer programs and/or modules. A software application may be part of the interoperable case series system, may be part of a larger system that includes the interoperable case series system, may be part of a separate system, or may not be a part of any system at all. A “case series” is a set of one or more adverse event cases. An “adverse event case” or “case” includes a data record that represents an adverse event. A “patient list” is a set of one or more medical records. A medical record is a data record that includes medical data. The interoperable case series system can include a central case series repository that can store one or more case series in addition to information related to the one or more case series, such as case series history information related to each case series, and case revision information related

to each case series. The interoperable case series system can further include an application programming interface ("API") that one or more software applications can interface with, so that the one or more software applications can produce or consume one or more case series using the case series repository. The interoperable case series can further include a case series data model that can contain a canonical representation of one or more case series, and information related to each case series, such as case series history information related to each case series, and case revision information related to each case series.

[0018] Fig. 1 illustrates a block diagram of a system 10 that can implement one embodiment of the invention. System 10 includes a bus 12 or other communications mechanism for communicating information between components of system 10. System 10 also includes a processor 22, operatively coupled to bus 12, for processing information and executing instructions or operations. Processor 22 may be any type of general or specific purpose processor. System 10 further includes a memory 14 for storing information and instructions to be executed by processor 22. Memory 14 can be comprised of any combination of random access memory ("RAM"), read only memory ("ROM"), static storage such as a magnetic or optical disk, or any other type of machine or computer-readable medium. System 10 further includes a communication device 20, such as a network interface card or other communications interface, to provide access to a network. As a result, a user may interface with system 10 directly, or remotely through a network or any other method.

[0019] A computer-readable medium may be any available medium that can be accessed by processor 22. A computer-readable medium may include both a volatile and nonvolatile medium, a removable and non-removable medium, a communication medium, and a storage medium. A communication medium may

include computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism, and may include any other form of information delivery medium known in the art. A storage medium may include RAM, flash memory, ROM, erasable programmable read-only memory ("EPROM"), electrically erasable programmable read-only memory ("EEPROM"), registers, hard disk, a removable disk, a compact disc read-only memory ("CD-ROM"), or any other form of storage medium known in the art.

[0020] Processor 22 can also be operatively coupled via bus 12 to a display 24, such as a Liquid Crystal Display ("LCD"). Display 24 can display information to the user. A keyboard 26 and a cursor control device 28, such as a computer mouse, can also be operatively coupled to bus 12 to enable the user to interface with system 10. In other embodiments, a user can interface with system 10 using a human interface device (not shown in Fig. 1), where a human interface device is a device configured to interact directly, and take input from, a user. Examples of a human interface device include a webcam, a fingerprint scanner, and a headset.

[0021] According to one embodiment, memory 14 can store software modules that may provide functionality when executed by processor 22. The modules can include an operating system 15, an interoperable case series module 16, as well as other functional modules 18. Operating system 15 can provide an operating system functionality for system 10. Interoperable case series module 16 can provide functionality for integrating a case series repository with applications that produce and consume case series, as will be described in more detail below. In certain embodiments, interoperable case series module 16 can comprise a plurality of modules, where each module provides specific individual functionality for integrating

a case series repository with applications that produce and consume case series. System 10 can also be part of a larger system. Thus, system 10 can include one or more additional functional modules 18 to include the additional functionality. For example, functional modules 18 may include modules that provide additional functionality, such as a module of the “Oracle Argus Insight” product from Oracle Corporation.

[0022] Processor 22 can also be operatively coupled via bus 12 to a database 34. Database 34 can store data in an integrated collection of logically-related records or files. Database 34 can be an operational database, an analytical database, a data warehouse, a distributed database, an end-user database, an external database, a navigational database, an in-memory database, a document-oriented database, a real-time database, a relational database, an object-oriented database, or any other database known in the art. Further, database 34 can be accessed through an API, and can support a query language.

[0023] Fig. 2 illustrates a block diagram of an interoperable case series system 200, according to an embodiment of the invention. Interoperable case series system 200 can include case series repository 210, case series data model 220, and case series API 230. In certain embodiments, interoperable case series system 200 can also include a user interface component (not illustrated in Fig. 2) that provides functionality for browsing and manipulating one or more case series.

[0024] According to the embodiment, case series repository 210 is a repository that can store data, such as one or more case series. For example, in an embodiment where a case series includes one or more case identifiers (where each identifier uniquely identifies each adverse event case included within the case series), and further includes case data and/or case metadata that together represent

the one or more adverse event cases included within the case series, case series repository 210 can store the one or more case identifiers, and the associated case data and/or case metadata. Furthermore, for each case series, case series repository 210 can also store information related to each case series, such as case series history information and case revision information, which is further described below in greater detail. Case series repository 210 can be any type of repository that can store data, such as a database or a file.

[0025] Further, according to the embodiment, case series data model 220 is a data model that can include a canonical representation of a case series, and information related to the case series. For example, in an embodiment where a case series includes one or more case identifiers and further includes case data and/or case metadata that together represent the one or more adverse event cases included within the case series, case series data model 220 can include a data field that represents the case identifier, and one or more additional data fields that represent the case data and/or case metadata.

[0026] Furthermore, where the case series includes information related to the case series, case series data model 220 can further include one or more additional data fields that represent the information related to the case series. In certain embodiments, the information related to the case series can include case series history information. Case series history information can include information regarding the history of the case series, such as the individual that generated the case series, the mechanism that generated the case series, the one or more individuals that have modified the case series, etc. In certain embodiments, case series history information can be stored in a format of one or more change logs that can be associated with a case series.

[0027] In other embodiments, the information related to the case series can include case revision information. Case revision information can include information regarding one or more revisions of the case series. According to an embodiment, a revision is any change to an adverse event case of the case series. Thus, according to the embodiment, a case series can include one or more case revisions. Further, each case revision can be identified by the partner application that created the case revision.

[0028] In addition to, or as an alternative to including information regarding one or more revisions of the case series, case revision information can include information regarding one or more versions of the case series. According to an embodiment, a version is a change to an adverse event case of the case series that has gone through a quality analysis cycle, and is identified as ready for scientific analysis. In other words, according to the embodiment, every version of a case series is also a revision of the case series, but not every revision of the case series is also a version of the case series. In one example, a producing software application may include in-process revisions to cases, but a consuming software application may only support work with final versions. Case series data model 220, in conjunction with case series API 230, can allow the software applications to interpret the case revision information so that it can be most accurately used in the consuming software application. Thus, a case series can also be identified as a case revision series. Further, case revision information can include multiple revisions of the same case. In certain embodiments, case revision information can be stored in a format of one or more revisions that can be associated with a case series. An example of case series data model 220 is further described in relation to Fig. 3.

[0029] According to the embodiment, case series API 230 is an API that can

expose case series data model 220, and that can represent case series data model 220 to a software application based on a format specified by the software application. Thus, case series API 230 can allow a software application to interface with case series repository 210. According to an embodiment, case series API 230 can provide functionality for producing, consuming, searching for and/or updating one or more case series. Further, in one embodiment, a first software application can produce a case series and store the case series in case series repository 210 using case series API 230. According to the embodiment, a second software application can consume the case series produced by the first software application and stored within case series repository 210 using case series API 230. Thus, rather than requiring the first software application to export the case series, and requiring the second software application to import the case series, the two software applications can interface with case series repository 210 using case series API 230.

[0030] According to an embodiment, case series repository 210, case series data model 220, and case series API 230 support four kinds of case series: (1) named case series; (2) active user case series; (3) single-use case series; and (4) case hit list. A named case series is a case series that includes an explicit unique name.

The name can be given to the named case series by a user or by a function or executable process of a producing software application. Further, a software application can request a named case series by name. Case series repository 210 can support search or browse functions on a named case series. An active case series is a case series that is associated with a user. An active case series can be named for the user that the active case series is associated with. Content of an active case series can exist until it is overwritten. Thus, an active case series can allow for the creation of a personal work space that can span multiple software

applications. A single-use case series is a case series that can exist within case series repository 210 for the single purpose of executing a single report. After a transaction completes, a single-use case series can be deleted. Case series repository 210 can orchestrate the execution of the report through an API that is separate from case series API 230. A case hit list is a case series that can be completely managed by a producing software application. When a case hit list is stored in case series repository 210, the case hit list can be given an identity known only to the producing software application. A case hit list does not appear in a list of named case series, but can be accessible to a consumer who is passed the identity by the producing software application.

[0031] Additionally, in one embodiment, case series repository 210, case series data model 220, and case series API 230 support other series that are very similar to case series: (1) event series; and (2) product series. A query for a particular adverse event that can be executed on a data source, such as a drug safety system, can produce a case series that can be stored within case series repository 210. Each case in the case series can have at least one event that matches the query. However, each case may also include additional events that do not match the query. Thus, a case series can include all cases that match a query, and all of the events related to those cases, whether the event matches the query or not. In contrast, an event series that is produced from a query executed on a data source can include all cases that match the query, but only include the events related to those cases that match the query as well. Further, a query for a particular medicinal or pharmaceutical product can also produce a case series. Each case in the case series can have at least one product that matches the query. However, each case may also include additional products that do not match the query. Thus, a case

series can include all cases that match a query, and all of the products related to those cases, whether the product matches the query or not. In contrast, a product series that is produced from a query executed on a data source can include all cases that match the query, but only include the products related to those cases that match the query as well. Additionally, named event series, named product series, active event series, active product series, single-use event series, single-use product series, event hit lists and product hit lists are valid variations, and can work in an analogous way, to the named case series, active case series, single-use case series, and case hit lists, previously described above.

[0032] According to an embodiment, case series API 230 can execute the following functions on a case series, event series, or product series: (a) view a series; (b) save a series; (c) make a series active; (d) assign access rights to a series; (e) add a case to a series; (f) delete a case from a series; (g) delete a series; (h) annotate a case; (i) annotate a series; (j) export a series; (k) freeze a series; and (l) merge two series. The aforementioned functionality is further described in greater detail.

[0033] A user can view information stored in a series including one or more case identifiers, case revision information, case series history information, any other information related to the series, a query criteria that created the series, or any other properties and/or metadata of the series. Viewing can include searching and sorting functionality. A user can also save a series and give it a name, making it a named series. A user can make a series into an active series. A user who created a series can assign access rights to the series, such as read access and/or write access. Additionally, a user can add a case to a series. A user can also delete a case from a series. Further, a user can delete a series from case series repository 210. Also, a

user can make a text annotation to a case in a series in the context of that series. If the same case appears in other series, the annotations for the case in the various series can be separate. A user can also make text annotations at a series level. Additionally, a user can export a series to a file. Further, a user can freeze a series, where the case data in the series does not change after the date and time it was frozen, even if the cases are updated in a corresponding data source that includes the case data, such as a drug safety system. A user can also unfreeze a series, so that the case data can again reflect the most current revisions available. A user can also merge two series using a union, intersect or minus operation, and thereby create a new series.

[0034] Case series API 230 can further perform provide the functionality to produce, consume, search, and update functions, according to the embodiment. In performing a produce function, case series API 230 can receive a series and store the series within case series repository 210. In performing a consume function, case series API 230 can retrieve a series from case series repository 210 and implement the series within a software application (such as displaying the series within a user interface of the software application). In performing a search function, case series API 230 can search for a series stored within case series repository 210. In performing an update function, case series API 230 can update a series stored within case series repository 210.

[0035] In certain embodiments, an embodiment can add new case series to case series repository 210 by executing a query on a data source, such as a drug safety system, where one or more cases returned by the query can be stored within case series repository 210 as a case series. Additionally, in some of these embodiments, a user can add new series to case series repository store in other ways than be

executing a query on a data source. More specifically, a user can add a new series by: (a) entering a series; or (b) importing a series. By entering a series, a user can manually enter one or more case identifiers within case series repository 210 to create one or more new series. Alternately, a user can import a new series into case series repository 210 from a file containing one or more case identifiers.

[0036] Fig. 3 illustrates a block diagram of a case series data model 300, according to an embodiment of the invention. In certain embodiments, case series data model 300 is identical to case series data model 220 of Fig. 2. As previously described, case series data model 300 includes a canonical representation of one or more case series, and information related to each case series, such as case series history information related to each case series, and case revision information related to each case series. As described below in greater detail, case series data model 300 can include a plurality of data fields, where each data field can represent a data field of a case series repository (such as case series repository 210 of Fig. 2), and where each data field can be represented in its own unique format by a case series API (such as case series API 230 of Fig. 2).

[0037] According to the embodiment, case series data model 300 includes case series 310. Case series 310 is a canonical representation of one or more case series. In certain embodiments, case series 310 includes a plurality of data fields, where each data field can store case data or metadata of each case series of the one or more case series. In some of these embodiments, case series 310 includes a data field that can store a case identifier of a case of the case series, and includes one or more data fields that can store case data or metadata that represent the one or more adverse event cases included within the case series. Thus, case series 310 can store a case identifier for each case of the case series, and case series 310 can

store the data and/or metadata related to each case of the cases series. Thus, in these embodiments, case series 310 can represent one or more case series by storing a plurality of values within a plurality of data fields.

[0038] Case series data model 300 also includes change log 320. Change log 320 is a canonical representation of case series history information that is related to a case series. As previously described, case series history information can include information regarding the history of a case series, such as the individual that generated the case series, the mechanism that generated the case series, the one or more individuals that have modified the case series, etc. In certain embodiments, change log 320 includes one or more data fields, where each data field can store case series history information of each case series of the one or more case series. According to these embodiments, each data field can store a value that represents a distinct component of the case series history information. For example, a first data field of change log 320 can store a name of an individual that generated a case series, a second data field can store a name of a mechanism that generated the case series, a third data field can store a name of a first individual that modified the case series, a fourth data field can store a name of a second individual that modified the case series, etc. According to an embodiment, case series 310 and change log 320 can have a one-to-many relationship, where one or more change logs can be associated with a case series. In one embodiment, change log 320 can be implemented as a character large object ("CLOB"), where each value of change log 320 can be appended to the end of the CLOB.

[0039] Case series data model 300 also includes case revision 330. Case revision 330 is a canonical representation of case revision information that is related to one or more case series. More specifically, in certain embodiments, a case series

is a container that contains one or more case revisions. In some scenarios, a case series can contain multiple case revisions of the same case. In other scenarios, a case series contains one case revision for each case of the case series. As previously described, case revision information includes information regarding one or more revisions of a case series and/or one or more versions of the case series, where a revision can be any change to an adverse event case of the case series, and a version can be a change to an adverse event case of the case series that has been verified according to a defined review process. In certain embodiments, case revision 330 includes one or more data fields, where each data field can store case revision information of each case series of the one or more case series. According to these embodiments, each data field can store a value that represents a distinct component of the case revision information. In certain embodiments, a revision and/or version of a case series can be represented in a format that is identical to a format of the original case series. Thus, in these embodiments, case revision 330 includes a data field that can store a case identifier of a case of the case series, and includes one or more data fields that can store case data or metadata of the case of the case series, where the case data or metadata includes the change to the adverse event case of the case series. In other embodiments, a revision and/or version of a case series can be represented in a format that solely includes the change to the adverse event case of the case series. Thus, in these embodiments, case revision 330 includes one or more data fields that store the change to the adverse event case of the case series. According to an embodiment, case series 310 and case revision 330 can have a one-to-many relationship, where one or more case revisions can be associated with a case series. In certain embodiments, case revision 330 also includes timestamp information that can be represented by two

additional data fields, a valid start date/time data field, and a valid end date/time data field. The time stamp information can represent a starting date and/or time and an ending date and/or time that the revision or version of the case series is valid.

Further, a valid start date/time and a valid end date/time, are both features of source data that can enable production of a case series with case revision information. The time stamp information can, either in whole or in part, identify the revision or version of the case series.

[0040] Case series data model 300 also includes annotation 340. Annotation 340 is a canonical representation of annotation information that is related to one or more case revisions of a case series. Annotation information can include any user-defined information, where the user-defined information can serve to annotate a case revision of a case series. In certain embodiments, annotation 340 includes one or more data fields, where each data field can store a user-defined value. According to an embodiment, case revision 330 and annotation 340 can have a one-to-many relationship, where one or more annotations can be associated with a case revision.

[0041] Case series data model 300 also includes folder 350. Folder 350 is a canonical representation of a logical grouping of one or more case series. Over time, a user can generate thousands of case series using case series data model 300. A nested folder storage system can be used to organize the case series. Thus, one or more case series can be associated with a folder, and one or more folders can be nested within a folder. Thus, according to an embodiment, folder 350 and case series 310 can have a one-to-many relationship, where one or more case series can be associated with a folder. In one embodiment, folder 350 is not part of case series data model 300, but instead is a representation of a folder functionality of a document management system that is leveraged in order to organize one or more

case series generated using case series data model 300 into one or more folders.

[0042] Fig. 4 illustrates a block diagram of an example implementation of an interoperable case series system, according to an embodiment of the invention. More specifically, Fig. 4 illustrates an example of an interoperable case series system, such as interoperable case series system 200 of Fig. 2, interacting with a plurality of software applications. The implementation includes case series repository 400. As previously described, case series repository 400 is a repository that can store data, such as one or more case series, and/or information related to the one or more case series. In certain embodiments, case series repository 400 is identical to case series repository 210 of Fig. 2. The implementation further includes case series data model 410. As also previously described, case series data model 410 is a data model that can include a canonical representation of a case series, and information related to the case series. In certain embodiments, case series data model 410 is identical to case series data model 220 of Fig. 2 and case series data model 300 of Fig. 3. According to the embodiment, case series data model 410 can be a data model that represents the data stored within case series repository 400. In one embodiment, case series data model 410 can include a plurality of data fields, where the plurality of data fields represents the plurality of data fields included within case series repository 400.

[0043] The implementation further includes data mining application 420 and data mining application case series API 430. According to the embodiment, data mining application 420 is a software application that includes one or more executable processes that can execute data mining functionality to find one or more related groups of cases within drug safety data. Data mining application 420 can produce one or more case series using one or more data mining algorithms. Data mining

application 420 can further consume one or more case series produced by another software application. In one embodiment, data mining application is an “Empirica Signal” product from Oracle Corporation.

[0044] Also according to the embodiment, data mining application case series API 430 provides an interface that exposes case series data model 410 to data mining application 420, that represents case series data model 410 to data mining application 420 based on a format specified by data mining application 420, and, thus, that allows data mining application 420 to interface with case series repository 400. Therefore, data mining application 420 can produce one or more case series, and can store the one or more produced case series within case series repository 400, using data mining application case series API 430. Likewise, data mining application 420 can retrieve one or more case series from within case series repository 400, and can consume the one or more retrieved case series, using data mining application case series API 430. In certain embodiments, data mining application case series API 430 represents a component of case series API 230 of Fig. 2.

[0045] The implementation further includes reporting application 440 and reporting application case series API 450. According to the embodiment, reporting application 440 is a software application that includes one or more executable processes that can execute reporting functionality to generate one or more reports that visualize one or more case series. Reporting application 440 can produce one or more case series using one or more reporting algorithms. Reporting application 440 can further consume one or more case series produced by another software application. In one embodiment, reporting application 440 is an “Oracle Argus Insight” product from Oracle Corporation.

[0046] Also according to the embodiment, reporting application case series API 450 provides an interface that exposes case series data model 410 to reporting application 440, that represents case series data model 410 to reporting application 440 based on a format specified by reporting application 440, and, thus, that allows reporting application 440 to interface with case series repository 400. Therefore, reporting application 440 can produce one or more case series, and can store the one or more produced case series within case series repository 400, using reporting application case series API 450. Likewise, reporting application 440 can retrieve one or more case series from within case series repository 400, and can consume the one or more retrieved case series, using reporting application case series API 450. Further, reporting application case series API 450 can simplify the use of case series in a report of reporting application 440, can allow a report of reporting application 440 to execute a query and use the resulting case series, and, if desired, can store the resulting case series in case series repository 400 for further use. In certain embodiments, reporting application case series API 450 represents a component of case series API 230 of Fig. 2.

[0047] Thus, according to the embodiment, data mining application 420 can interact with reporting application 440, and vice-versa, as reporting application 440 can access one or more case series produced by data mining application 420, and data mining application 420 can access one or more case series produced by reporting application 440. One of ordinary skill in the art would readily appreciate that data mining application 420 and reporting application 440 are examples of software applications that produce and consume case series according to the embodiment, and that, in alternate embodiments, data mining application 420 and reporting application 440 can be replaced with other software applications that

include alternate functionality. Further, there can be any number of case series APIs that support any number of software applications, and that can allow any number of software applications to access case series repository 400 using case series data model 410.

[0048] Fig. 5 illustrates a flow diagram of the functionality of an interoperable case series module, according to an embodiment of the invention. In one embodiment, the functionality of the flow diagram of Fig. 5 (described below), as well as the functionality of the flow diagram of Fig. 8 (also described below), are each implemented by software stored in a memory or some other computer-readable or tangible medium, and executed by a processor. In other embodiments, each functionality may be performed by hardware (e.g., through the use of an application specific integrated circuit ("ASIC"), a programmable gate array ("PGA"), a field programmable gate array ("FPGA"), etc.), or any combination of hardware and software.

[0049] The flow begins and proceeds to 510. At 510 a case series is received. In certain embodiments, the case series is received from a partner application. The case series includes one or more adverse event cases, and each adverse event case includes a data record that represents an adverse event. The case series can be a named case series, an active user case series, a single-use case series, or a case hit list. The case series can further be an event series or a product series. In certain embodiments, each data record that represents an adverse event further includes drug safety data, where drug safety data includes one or more reports or patient identifiers that are related to the safety of one or more drugs. In certain embodiments, the case series can include a plurality of case revisions. The flow proceeds to 520.

[0050] At 520, information related to the case series is received. In certain embodiments, the information related to the case series is received from a partner application. In certain embodiments, the information related to the case series includes case revision information, where case revision information can include at least one change to an adverse event case of the case series. In these embodiments, a case revision is received that includes the case revision information. In certain embodiments, where the at least one change to the adverse event case of the case series is verified according to a defined review process, a case version is received that includes the case revision information. Further, in certain embodiments, the case revision information includes timestamp information that identifies the case revision. The timestamp information can include a valid start date and/or time and a valid end date and/or time.

[0051] In other embodiments, the information related to the case series includes case series history information, where case series history information can include information regarding the history of the case series, such as an individual that generated the case series, a mechanism that generated the case series, or one or more individuals that have modified the case series. In these embodiments, a change log is created that includes the case series history information. In other embodiments, the information related to the case series includes user-defined information. In these embodiments, an annotation is created that includes the user-defined information. In other embodiments, the information related to the case series includes a logical organization of the case series and one or more additional case series. In these embodiments, a folder is created that includes the logical organization of the case series and the one or more additional case series. The flow proceeds to 530.

[0052] At 530, the case series, and the information related to the case series, are stored within a case series repository using a case series data model. A case series repository is a repository that can store data, such as one or more case series and information related to the one or more case series. A case series data model is a canonical representation of a case series that defines a format of the case series and the information related to the case series within the case series repository.

[0053] In embodiments where the information related to the case series includes case revision information, the case series data model defines a format of the case revision within the case series repository. In embodiments where the information related to the case series includes case series history information, the case series data model defines a format of the change log within the case series repository. In embodiments where the information related to the case series includes user-defined information, the case series data model defines a format of the annotation within the case series repository. In embodiments where the information related to the case series includes a logical organization of the case series and one or more additional case series, the case series data model defines a format of the folder within the case series repository.

[0054] In certain embodiments, the case series data model includes a data field that represents one or more case identifiers of the case series, and one or more additional data fields that represent the one or more adverse event cases of the case series. In these embodiments, each case identifier uniquely identifies an adverse event case of the one or more adverse event cases. Further, in some of these embodiments, the case series data model includes one or more additional fields that represent the information related to the case series.

[0055] In embodiments where the information related to the case series includes

case revision information, the case series data model includes one or more additional data fields that represent the case revision information. In embodiments where the case revision information includes timestamp information, the case series data model includes one or more additional data fields that represent the timestamp information. In embodiments where the information related to the case series includes case series history information, the case series data model includes one or more additional data fields that represent the case series history information. In embodiments where the information related to the case series includes user-defined information, the case series data model includes one or more additional data fields that represent the user-defined information. In embodiments where the information related to the case series includes a logical organization of the case series and one or more additional case series, the case series data model the case series data model includes one or more additional data fields that represent the logical organization of the case series and one or more additional case series. The flow proceeds to 540.

[0056] At 540, the information related to the case series is associated within the case series using the case series data model. In embodiments where the information related to the case series includes case revision information, the case revision is associated with the case series using the case series data model. In embodiments where the information related to the case series includes case series history information, the change log is associated with the case series using the case series data model. In embodiments where the information related to the case series includes user-defined information, the annotation is associated with the case revision using the case series data model. In embodiments where the information related to the case series includes a logical organization of the case series and one or more

additional case series, the case series is associated with the folder. The flow proceeds to 550.

[0057] At 550, the case series and the associated information related to the case series is retrieved from the case series repository using a case series API. The case series API is an API that represents the case series data model to a software application based on a format specified by the software application. Thus, the case series API can define a format of the case series and the information related to the case series for the software application. The case series API can use the case series data model to retrieve the case series and the associated information from the case series repository. The flow then ends.

[0058] Fig. 6 illustrates a block diagram of a cohort identification system 600, according to an embodiment of the invention. Components of cohort identification system 600 that are shaded in Fig. 6 are components that can change depending on a data model of a data source, as will be described below in greater detail. Cohort identification system 600 can include query builder user interface ("UI") 605. Query builder UI 605 is a user interface that can be displayed to a user of cohort identification system 600, where query builder UI 605 can allow a user to create a query. In other embodiments, query builder UI 605 can display one or more data fields of a data source, and a user can select at least one of the one or more data fields to be part of the query. In these embodiments, a user can also enter criteria that can be part of the query. As will be described in greater detail, the query created by the user can be executed on a data source, such as a drug safety system, in order to retrieve data stored within the data source, such as drug safety data. In certain embodiments, a user can enter SQL syntax for the query within query builder UI 605. Further, in certain embodiments, query builder UI 605 can

allow an author of the query to specify one or more place holders, identified as parameters. When the query is executed, a user can be prompted to enter a parameter value for each parameter. In addition, query builder UI 605 can also allow a user to execute a query.

[0059] Cohort identification system 600 can also include metadata 610.

According to the embodiment, metadata 610 describes the data within a data source, such as drug safety data within a drug safety system. More specifically, metadata 610 describes information about each data field of the data source that can be queried. Such information can include a data type of the data field and information required to construct a SQL query that include the data field. Metadata 610 can also include one or more query fields that can be derived from source data, or a combination of source data and reference data. According to the embodiment, query builder UI 605 can retrieve metadata 610 so that a user can create a query based on metadata 610 using query builder UI 605. Because of metadata 610, cohort identification system 600 is not limited to only be operatively coupled to a particular data source, or a particular data model of a data source. Instead, cohort identification system 600 is data model-independent, and can be operatively coupled with a wide variety of data sources, as will be described in greater detail. Metadata 610 can be stored in any data structure contained within cohort identification system 600, such as a repository.

[0060] Cohort identification system 600 can further include query repository 615. Query repository 615 is a repository that can store one or more queries. According to the embodiment, query builder UI 605 can store a query that is created within query builder UI 605 within query repository 615. A query created within query builder UI 605 can be stored within query repository 615 when it is determined that

the query can likely be subsequently reused, such as when the query can retrieve data that will likely be used in a wide range of scenarios.

[0061] Cohort identification system 600 can also include query compiler 620. Query compiler 620 can retrieve a query that is stored within query repository 615, and can compile the stored query. By compiling the stored query, query compiler 620 can convert the query into an executable format, so that the stored query can be compiled. In some embodiments, query compiler 620 can further execute the query, once the query has been converted into an executable format. In executing the query, query compiler 620 can execute the query on a data source, and can retrieve and store data that is returned by the data source based on the query. In some embodiments, the data that is returned by the data sources includes drug safety data, where the drug safety data includes one or more adverse event cases, where each adverse event case is a data record that represents an adverse event. Further, the execution of the query can create a case series.

[0062] Cohort identification system 600 can further include case series repository 625. Case series repository 625 is a repository that can store data, such as one or more case series. For example, in an embodiment where a case series includes one or more case identifiers and further includes case data and/or case metadata that together represent the one or more adverse event cases included within the case series, case series repository 625 can store the one or more case identifiers, and the associated case data and/or case metadata. According to the embodiment, once query compiler 620 executes a query and creates a case series, query compiler can store the created case series within case series repository 625. In certain embodiments, case series repository 625 can include an associated case series data model (not illustrated in Fig. 6) that can include a canonical representation of a case

series. For example, in an embodiment where a case series includes one or more case identifiers and further includes case data and/or case metadata that together represent the one or more adverse event cases included within the case series, the case series data model can include a data field that represents the case identifier, and one or more additional data fields that represent the case data and/or case metadata. In certain embodiments, case series repository 625 is identical to case series repository 210 of Fig. 2, and case series repository 400 of Fig. 4.

[0063] Cohort identification system 600 can also include reporting case series API 630. According to the embodiment, reporting case series API 630 is an API that can expose the case series data model associated with case series repository 625, and that can represent the case series data model associated with case series repository 625 to a reporting application (such as reporting application 640) based on a format specified by the reporting application. Thus, reporting case series API 630 can allow a reporting application (such as reporting application 640) to interface with case series repository 625. In other words, reporting case series API 630 can retrieve a case series from case series repository 625 and implement the series within a reporting application (such as reporting application 640). In certain embodiments, reporting case series API 630 represents a portion of case series API 230 of Fig. 2, and is identical to reporting application case series API 450 of Fig. 4. Reporting application 640 is a software application that includes one or more executable processes that can execute reporting functionality to generate one or more reports that visualize one or more case series. The generating the one or more reports can include displaying the one or more reports within reporting application 640. Reporting application 640 can also produce one or more case series using one or more reporting algorithms. Reporting application 640 can further

consume one or more case series produced by cohort identification system 600 that are stored within case series repository 625 using reporting case series API 630. In certain embodiments, reporting application 640 is identical to reporting application 440 of Fig. 4.

[0064] Cohort identification system 600 can further include interoperable case series API 635. According to the embodiment, interoperable case series API 635 is an API that can expose the case series data model associated with case series repository 625, and that can represent the case series data model associated with case series repository 625 to a partner application (such as partner application 650) based on a format specified by the partner application. Thus, interoperable case series API 635 can allow a partner application (such as partner application 650) to interface with case series repository 625. In other words, interoperable case series API 635 can retrieve a case series from case series repository 625 and implement the series within a partner application (such as partner application 650). Partner application 650 is a software application that can consume one or more case series produced by cohort identification system 600 that are stored within case series repository 625 using interoperable case series API 635. Partner application 650 can also provide other functionality, such as creating one or more cases series that can be stored within case series repository 625 using interoperable case series API 635. In certain embodiments, interoperable case series API 635 is identical to case series API 230 of Fig. 2.

[0065] Cohort identification system 600 can also include compiler rules 645. According to the embodiment, compiler rules 645 can include one or more syntax rules that can be applied, by query compiler 620, to a query created by query builder UI 605 in order to determine that the query complies with the one or more syntax

rules. Compiler rules 645 can be stored in any data structure contained within cohort identification system 600, such as a repository.

[0066] Cohort identification system 600 can further include ontology browser UI 655. In embodiments where a data source is a reference ontologies data source, where a reference ontologies data source is described below in greater detail, ontology browser UI 655 can retrieve one or more reference ontologies from the reference ontologies data source and display the one or more reference ontologies to a user of cohort identification system 600 within a UI. Thus, one or more elements from an ontology can be selected and used as criteria in a query.

[0067] Cohort identification system 600 can further include case series editor and management UI 665. Case series editor and management UI 665 can allow a user of cohort identification system 600 to edit and manage one or more case series stored within case series repository 625.

[0068] Cohort identification system 600 can also include case series viewer UI 675. Case series viewer UI 675 can allow a user of cohort identification system 600 to view one or more case series. The one or more case series can be stored within case series repository 625. Alternatively, the one or more case series can be stored within a data source.

[0069] Further, according to the embodiment, cohort identification system 600 can be operatively coupled to one or more data sources. As previously described, components of cohort identification system 600 (i.e., query builder UI 605 and query compiler 620) can allow a user to create and execute one or more queries on one or more data sources operatively coupled to cohort identification system 600. In certain embodiments, the one or more data sources can include drug safety data, and in some of these embodiments, drug safety data can include data related to the safety

of one or more drugs, such as one or more adverse event cases. In the illustrated embodiment of Fig. 6, the one or more data sources include reference ontologies data source 660, and adverse event report databases 670, 680, and 690. Reference ontologies data source 660 is a data source that includes data regarding reference ontologies. Examples of reference ontologies data source 660 include a Systematized Nomenclature of Medicine ("SNOMED") data source, a Medical Dictionary for Regulatory Activities ("MedDRA") data source, or a World Health Organization ("WHO") drug data source. Adverse event report databases 678, 680, and 690 are data sources that include drug safety data, where the drug safety data includes one or more adverse event cases. However, these data sources are merely example data sources according to the illustrated embodiment, and in alternate embodiments, cohort identification system 600 can be operatively coupled to any number of data sources, and each data source can be any type of data source that includes data.

[0070] Cohort identification system 600 can further include federated query execution engine 685. Federated query execution engine 685 can allow a stored query to be compiled and be executed against multiple data sources. Federated query execution engine 685 can further merge the one or more case series returned from each data source into a single case series.

[0071] Cohort identification system 600 can also include flexible recategorization API 695. Flexible recategorization API 695 can normalize an interface to one or more code lists used in a data source. In most health related databases, discrete values are stored as codes. Code lists can be used to display one or more natural language equivalent terms. This feature can allow a user to specify query criteria in his, or her, own language. Further, one or more roll-up terms, such as "continent,"

can be used to refer to a group of discrete values, such as countries. Flexible recategorization API 695 can also allow one or more ranges of a continuous variable, such as age, to be mapped into one or more discrete named categories, such as “adult” or “child.” Flexible recategorization API 695 can allow the same code mapping to be used in reporting, thus, ensuring consistency between the query and the reports.

[0072] Fig. 7 illustrates a block diagram of an example implementation of a cohort identification system, according to an embodiment of the invention. At 710, a query is created. The query can be executed on a data source, such as a drug safety system, in order to retrieve data stored within the data source, such as drug safety data. According to the embodiment, in order to create the query, metadata can be retrieved, where the metadata describes the data within the data source. More specifically, the metadata can describe information about each data field of the data source that can be queried. Such information can include a data type of the data field and information required to construct a SQL query that include the data field. The query that is created is further stored at query repository 720, where query repository 720 is a repository that can store data, such as one or more queries.

[0073] At 730, a query is retrieved from query repository 720, where the query is compiled and executed on adverse event report database 740, an example of a data source. Adverse event report database 740 is a data source that includes drug safety data, where the drug safety data includes one or more adverse event cases. In executing the query, data, such as drug safety data, can be retrieved from adverse event report database 740. Further, in executing the query, a case series can be created and stored within case series repository 750.

[0074] At 760, the case series is retrieved from case series repository 750, and

reports 770 are generated that can visualize the case series. According to an embodiment, a reporting case series API can interface with case series repository 750 and retrieve the case series from case series repository 750 and implement the case series within a reporting application, in order to generate the one or more reports that can visualize the case series, where the reporting application can display the generated one or more reports. In certain embodiments, the generation of reports that is performed at 760, can include retrieving data from adverse event report database 740.

[0075] Fig. 8 illustrates a flow diagram of the functionality of a cohort identification module, according to an embodiment of the invention. The flow begins and proceeds to 810. The flow can begin when a user indicated that he, or she, wants to create a query. At 810, metadata is retrieved, where the metadata includes information about one or more data fields of a data source. According to the embodiment, the information can include a data type and SQL information for each data field of the one or more data fields. In certain embodiments, the data source can be an adverse event report database that stores one or more adverse event cases. The flow proceeds to 820.

[0076] At 820, a query is created for the data source based on the retrieved metadata. The query can be a query that is executed on a data source, in order to retrieve data stored within the data source. In certain embodiments, the retrieved metadata can be used to determine one or more data fields of the data source that are part of the query. Also, in certain embodiments, the retrieved metadata can be used to determine SQL that is part of the query. The flow proceeds to 830.

[0077] At 830, the query is compiled based on one or more compiler rules. According to the embodiment, compiler rules can include one or more syntax rules

that are applied to the query to determine that the query complies with the one or more syntax rules. In certain embodiments, the query can be stored in a query repository. The flow proceeds to 840.

[0078] At 840, the query is executed on the data source, where the execution of the query creates a case series. In certain embodiments, the case series includes one or more adverse event cases, where each adverse event case includes a data record that represents an adverse event. In some of these embodiments, each data record that represents an adverse event further includes drug safety data, where drug safety data includes one or more reports or patient identifiers that are related to the safety of one or more drugs. In certain embodiments, the case series is stored in a case series repository. The flow proceeds to 850.

[0079] At 850, a report is generated based on the case series, where the report is a visualization of the case series. In certain embodiments, the report includes a visual display of one or more data fields of the case series. According to certain embodiments, the case series can be retrieved from the case series repository using a reporting case series API, where the reporting case series API defines a format of the case series for a reporting application. In these embodiments, the report can be displayed within the reporting application. Also, in certain embodiments, the case series can be retrieved from the case series repository using an interoperable case series API, where the interoperable case series API defines a format of the case series for a partner application. In these embodiments, the case series can be consumed within the partner application. The flow then ends.

[0080] Fig. 9 illustrates an example query 910 that creates an example case series 920 that creates an example report 930, according to an embodiment of the invention. According to an embodiment, an executable process can execute query

910 on a data source, such as a drug safety system, where query 910 is a query to retrieve all fatal adverse event cases (i.e., all adverse event cases where a data field “Death” has a value of 1).

[0081] The execution of query 910 produces case series 920, according to the embodiment, where case series 920 comprises a list of adverse event cases that matches the conditions specified in query 910. Case series 920 can include at least a case identifier for each adverse event case, and may also include additional case data or metadata that represent the adverse event cases in the case series. In the illustrated embodiment, case series 920 includes a plurality of adverse event cases, where each adverse event case includes: (a) a case identifier data field, where each value identifies a case identifier for the adverse event case; and (b) a country data field, where each value identifies a country that the adverse event case is associated with.

[0082] According to an embodiment, an executable process can generate report 930 based on case series 920. Report 930 is a visualization of case series 920, where the data fields of case series 920 can be changed depending on a desired format. In the illustrated embodiment, report 930 includes a plurality of adverse event cases, where each adverse event case includes: (a) a case identifier data field, where each value identifies a case identifier for the adverse event case; (b) a “serious” data field, where each value identifies whether the adverse event case is a serious adverse event case; and (c) a “listed” data field, where each value identifies whether the adverse event case is a listed adverse event case. One of ordinary skill in the art would readily appreciate that the formats of query 910, case series 920, and report 930 are example formats according to an example embodiment, and that queries, case series, and/or reports can have other formats in alternate

embodiments.

[0083] Thus, in one embodiment, an interoperable case series system is provided that can facilitate a transfer of case series between multiple software applications. According to the embodiment, the interoperable case series system can enable a much higher degree of cross-application automation. Further, the interoperable case series can increase the scientific integrity and verifiability of conclusions reached using data, and can make it easier for individual to check and refine their work. The interoperable case series system can be relevant to any product in the drug safety space, because the interoperable case series system can be used to better integrate two or more applications that are required to share a case series, or to better integrate a transactional and reporting module of a single application.

[0084] Beyond drug safety, the interoperable case series system can be used in other health science domains such as epidemiology and translational medicine research. For example, a health science application can be used to identify a cohort of patients and save it as a patient list. The interoperable case series system can be used as an interface between a cohort identification/query application and a business intelligence visualization application. Anytime there is a need for these two applications to transfer a patient list to each other, the interoperable case series system could facilitate the transfer of the patient list. Further, many healthcare institutions have networks of applications and databases that contain data for the same patients. An interoperable case series system, according to an embodiment, could facilitate the transfer of a patient list between these applications.

[0085] The features, structures, or characteristics of the invention described throughout this specification may be combined in any suitable manner in one or more embodiments. For example, the usage of “one embodiment,” “some

embodiments,” “certain embodiment,” “certain embodiments,” or other similar language, throughout this specification refers to the fact that a particular feature, structure, or characteristic described in connection with the embodiment may be included in at least one embodiment of the present invention. Thus, appearances of the phrases “one embodiment,” “some embodiments,” “a certain embodiment,” “certain embodiments,” or other similar language, throughout this specification do not necessarily all refer to the same group of embodiments, and the described features, structures, or characteristics may be combined in any suitable manner in one or more embodiments.

[0086] One having ordinary skill in the art will readily understand that the invention as discussed above may be practiced with steps in a different order, and/or with elements in configurations which are different than those which are disclosed. Therefore, although the invention has been described based upon these preferred embodiments, it would be apparent to those of skill in the art that certain modifications, variations, and alternative constructions would be apparent, while remaining within the spirit and scope of the invention. In order to determine the metes and bounds of the invention, therefore, reference should be made to the appended claims.

WE CLAIM:

1. A computer-readable medium having instructions stored thereon that, when executed by a processor, cause the processor to integrate a case series repository with one or more software applications, the integrating comprising:

receiving a case series, wherein the case series comprises one or more adverse event cases, and wherein each adverse event case comprises a data record that represents an adverse event;

receiving a case revision, wherein the case revision comprises case revision information, and wherein the case revision information comprises at least one change to an adverse event case of the case series;

storing the case series and the case revision within a case series repository using a case series data model, wherein the case series data model defines a format of the case series and the case revision within the case series repository;

associating the case revision with the case series using the case series data model; and

retrieving the case series and the associated case revision using a case series application programming interface, wherein the case series application programming interface defines a format of the case series and the associated case revision for a software application.

2. The computer-readable medium of claim 1, wherein the case series data model comprises a data field that represents one or more case identifiers of the case series, and one or more additional data fields that represent the one or more adverse event cases of the case series, and wherein each case identifier uniquely identifies an adverse event case of the one or more adverse event cases.

3. The computer-readable medium of claim 2, wherein the case series data model comprises one or more additional data fields that represent the case revision information.

4. The computer-readable medium of claim 1, wherein the case revision information comprises timestamp information that identifies the case revision.

5. The computer-readable medium of claim 4, wherein the timestamp information comprises a valid start date and/or time and a valid end date and/or time; and

wherein the case series data model comprises one or more additional data fields that represent the timestamp information.

6. The computer-readable medium of claim 1, the integrating further comprising:

creating a change log, wherein the change log comprises case series history information, and wherein the case series history information comprises information regarding the history of the case series;

storing the change log within the case series repository using the case series data model, wherein the case series data model defines a format of the change log within the case series repository; and

associating the change log with the case series using the case series data model.

7. The computer-readable medium of claim 1, the integrating further comprising:

creating an annotation, wherein the annotation comprises user-defined information;

storing the annotation within the case series repository using the case series data model; and

associating the annotation with the case revision using the case series data model.

8. The computer-readable medium of claim 1, the integrating further comprising:

creating a folder, wherein the folder comprises a logical organization of the case series, and one or more additional case series;

storing the folder within the case series repository using the case series data model; and

associating the case series with the folder using the case series data model.

9. The computer-readable medium of claim 1, wherein the case series comprises one of: a named case series; an active user case series; a single-use case series; or a case hit list.

10. The computer-readable medium of claim 1, wherein the case series comprises one of: an event series; or a product series.

11. The computer-readable medium of claim 1, wherein the case series comprises one or more reports or patient identifiers that are related to the safety of one or more drugs.

12. The computer-readable medium of claim 1, wherein the case series comprises a plurality of case revisions.

13. A computer-implemented method for integrating a case series repository with one or more software applications, the computer-implemented method comprising:

receiving a case series, wherein the case series comprises one or more adverse event cases, and wherein each adverse event case comprises a data record that represents an adverse event;

receiving a case revision, wherein the case revision comprises case revision information, and wherein the case revision information comprises at least one change to an adverse event case of the case series;

storing the case series and the case revision within a case series repository using a case series data model, wherein the case series data model defines a format of the case series and the case revision within the case series repository;

associating the case revision with the case series using the case series data model; and

retrieving the case series and the associated case revision using a case series application programming interface, wherein the case series application programming interface defines a format of the case series and the associated case revision for a software application.

14. The computer-implemented method of claim 13, wherein the case series data model comprises a data field that represents one or more case identifiers of the case series, and one or more additional data fields that represent the one or more adverse event cases of the case series, and wherein each case identifier uniquely identifies an adverse event case of the one or more adverse event cases.

15. The computer-implemented method of claim 14 wherein the case series data model comprises one or more additional data fields that represent the case revision information.

16. The computer-implemented method of claim 13, wherein the case revision information comprises timestamp information that identifies the case revision.

17. The computer-implemented method of claim 16, wherein the timestamp information comprises a valid start date and/or time and a valid end date and/or time;
and

wherein the case series data model comprises one or more additional data fields that represent the timestamp information.

18. A system for integrating a case series repository with one or more software applications, the system comprising:

a processor;

a memory configured to store one or more instructions;

a case series reception module configured to receive a case series, wherein the case series comprises one or more adverse event cases, and wherein each adverse event case comprises a data record that represents an adverse event;

a case revision reception module configured to receive a case revision, wherein the case revision comprises case revision information, and wherein the case revision information comprises at least one change to an adverse event case of the case series;

a storage module configured to store the case series and the case revision within a case series repository using a case series data model, wherein the case series data model defines a format of the case series and the case revision within the case series repository;

an association module configured to associate the case revision with the case series using the case series data model; and

a retrieval module configured to retrieve the case series and the associated case revision using a case series application programming interface, wherein the case series application programming interface defines a format of the case series and the associated case revision for a software application.

19. The system of claim 18, wherein the case series data model comprises a data field that represents one or more case identifiers of the case series, and one or more additional data fields that represent the one or more adverse event cases of the case series, and wherein each case identifier uniquely identifies an adverse event case of the one or more adverse event cases.

20. The system of claim 19 wherein the case series data model comprises one or more additional data fields that represent the case revision information.

21. The system of claim 18, wherein the case revision information comprises timestamp information that identifies the case revision.

22. The system of claim 21, wherein the timestamp information comprises a valid start date and/or time and a valid end date and/or time; and

wherein the case series data model comprises one or more additional data fields that represent the timestamp information.

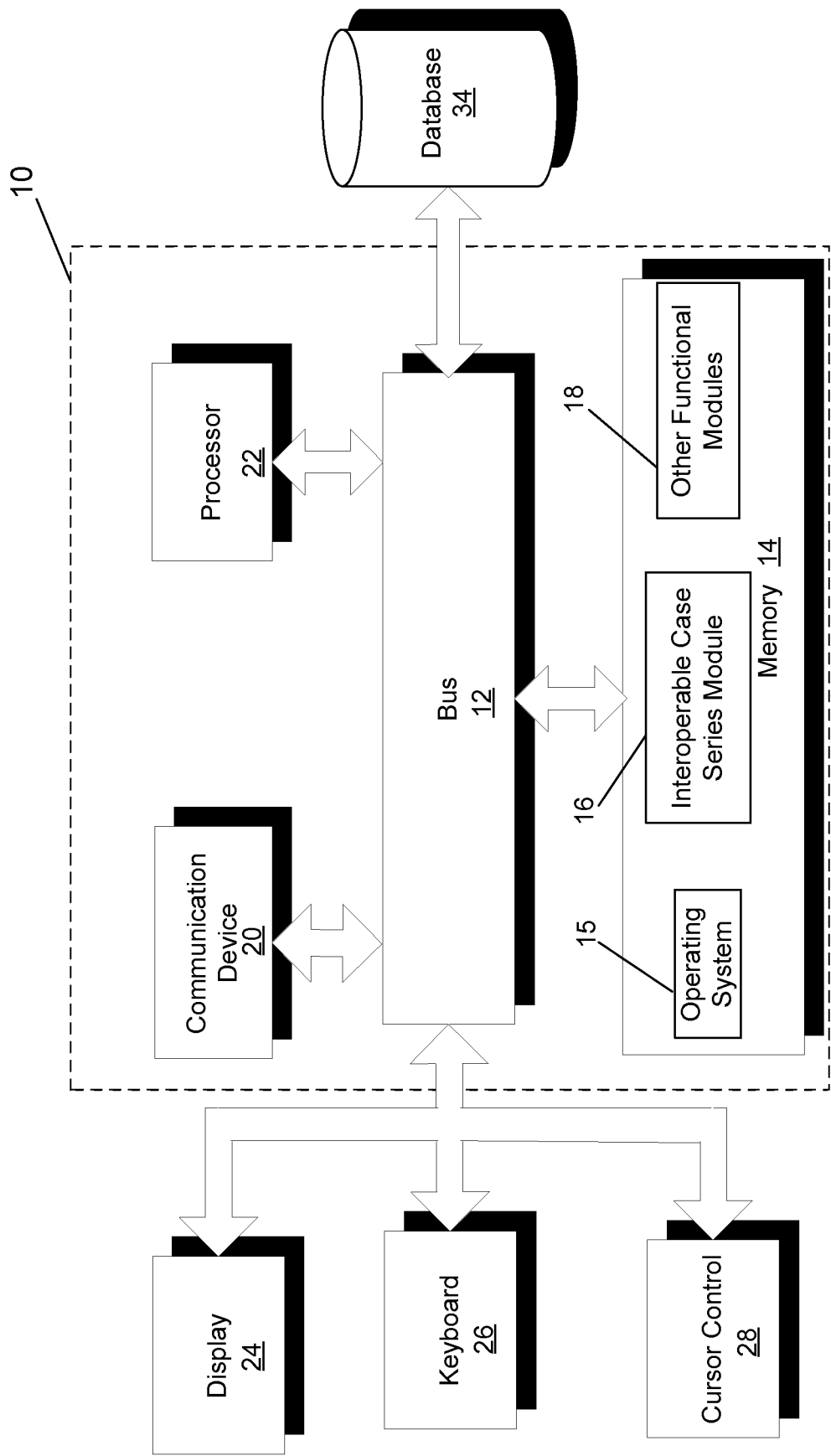


Fig. 1

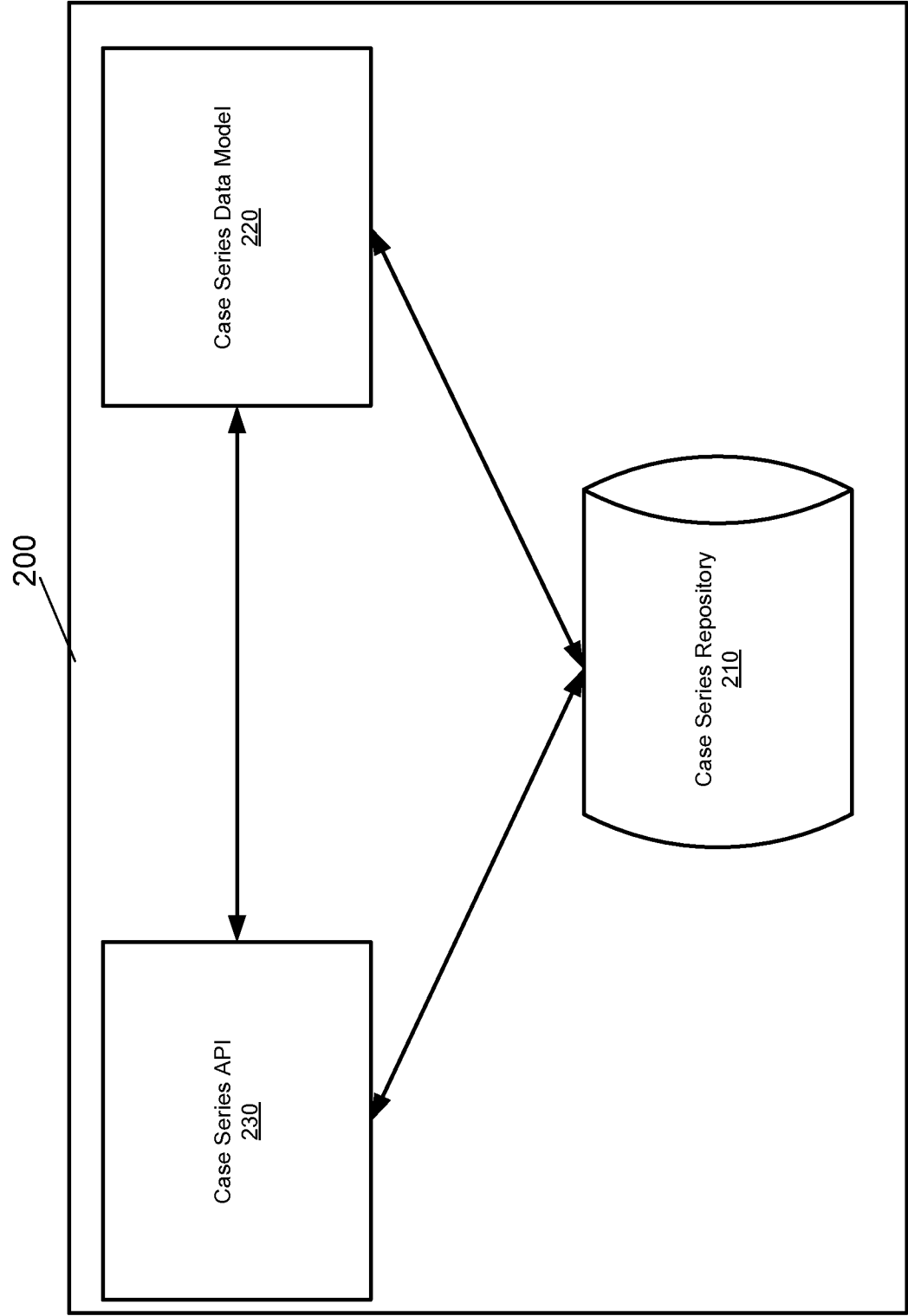


Fig. 2

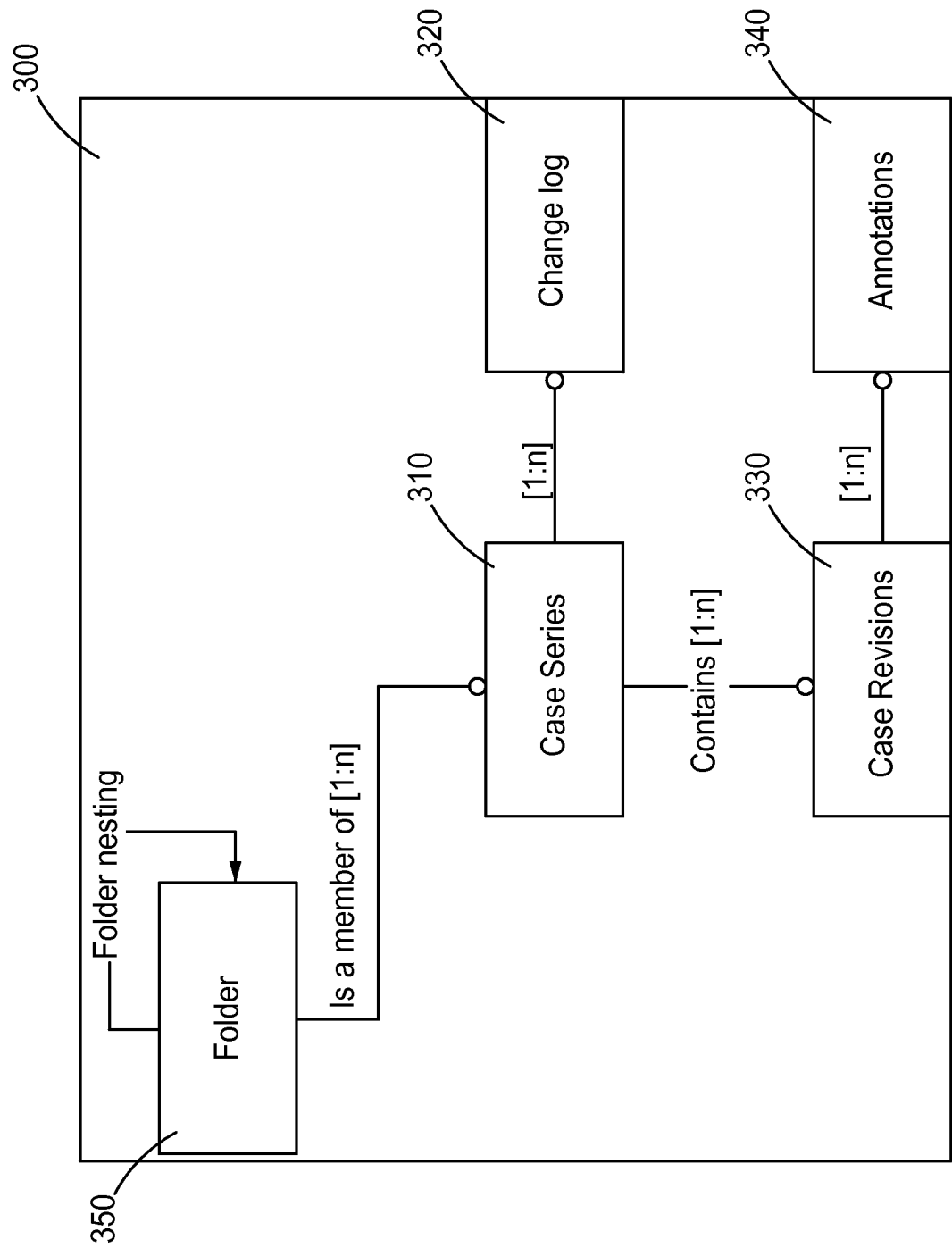


Fig.3

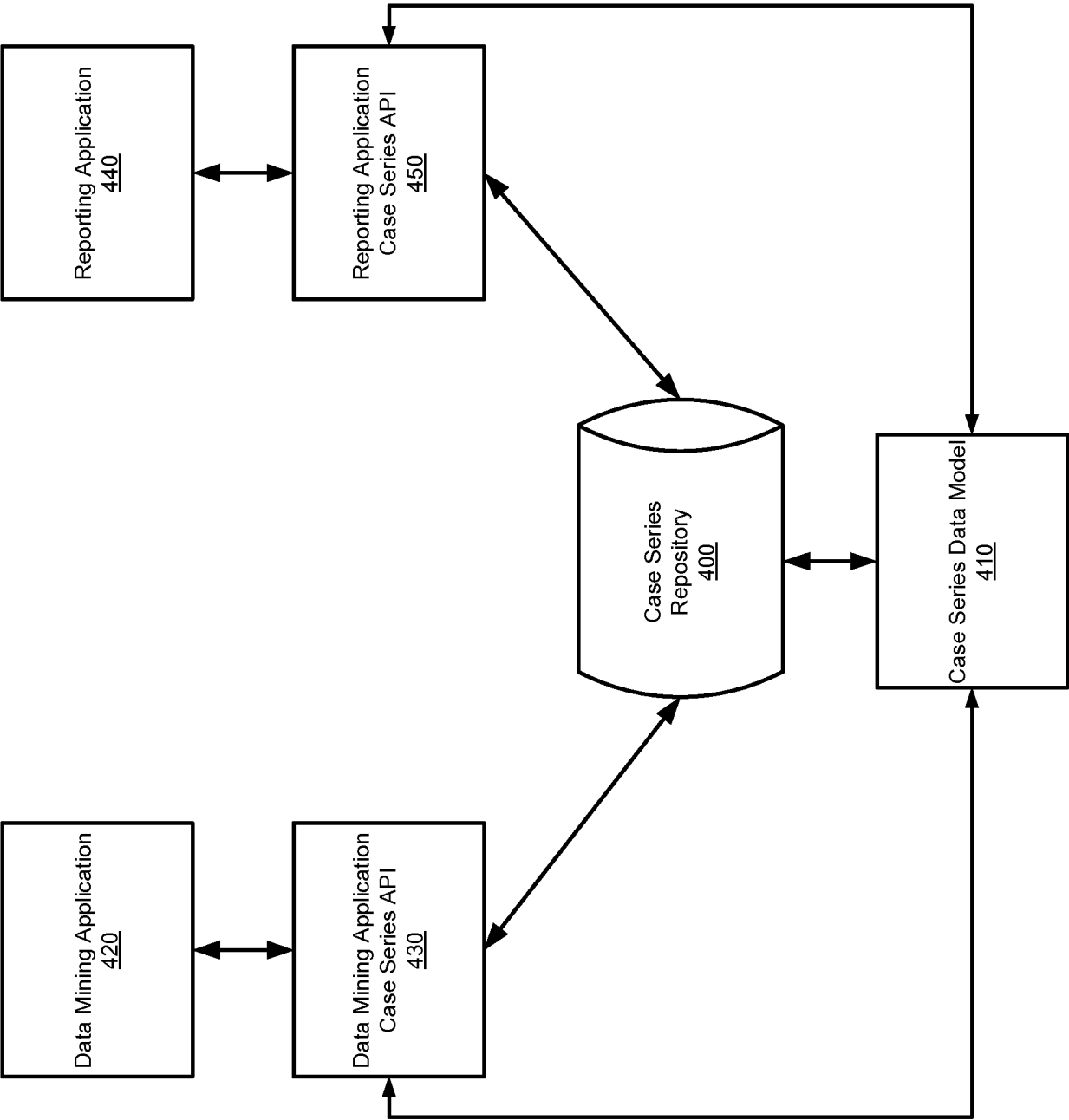


Fig. 4

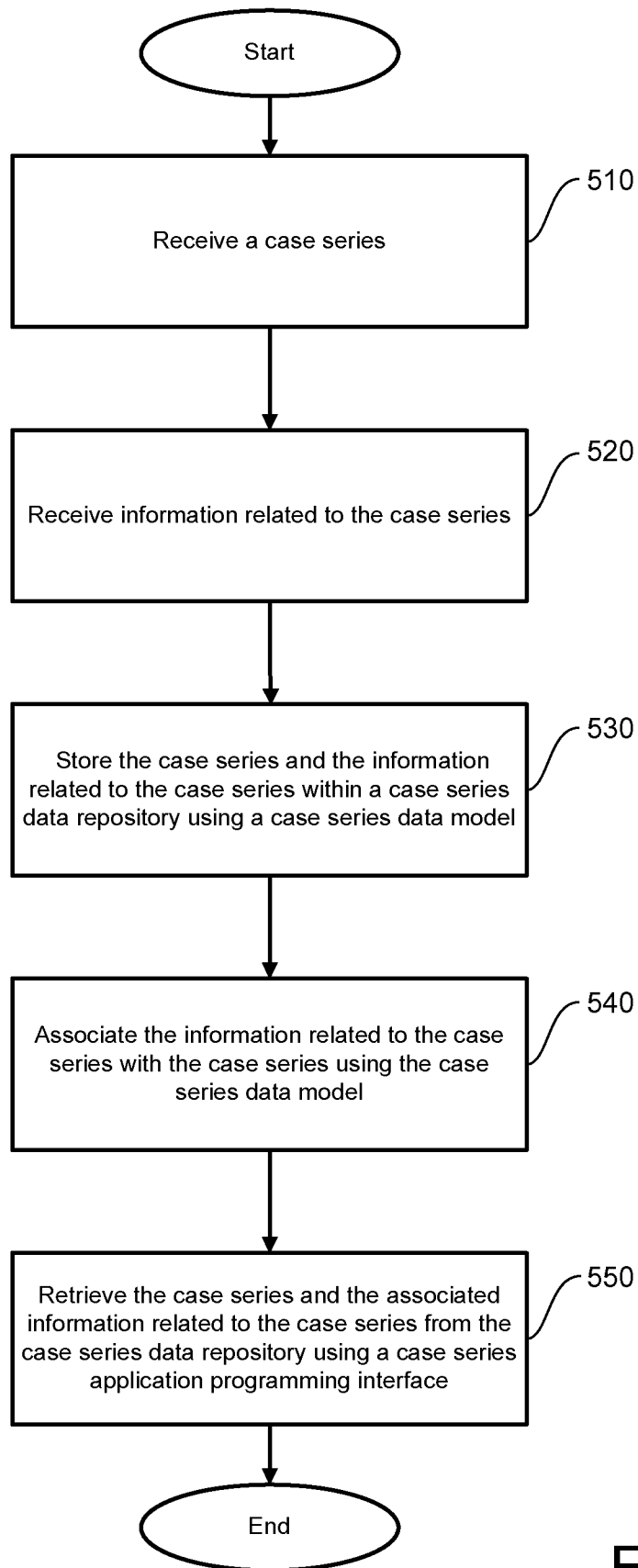
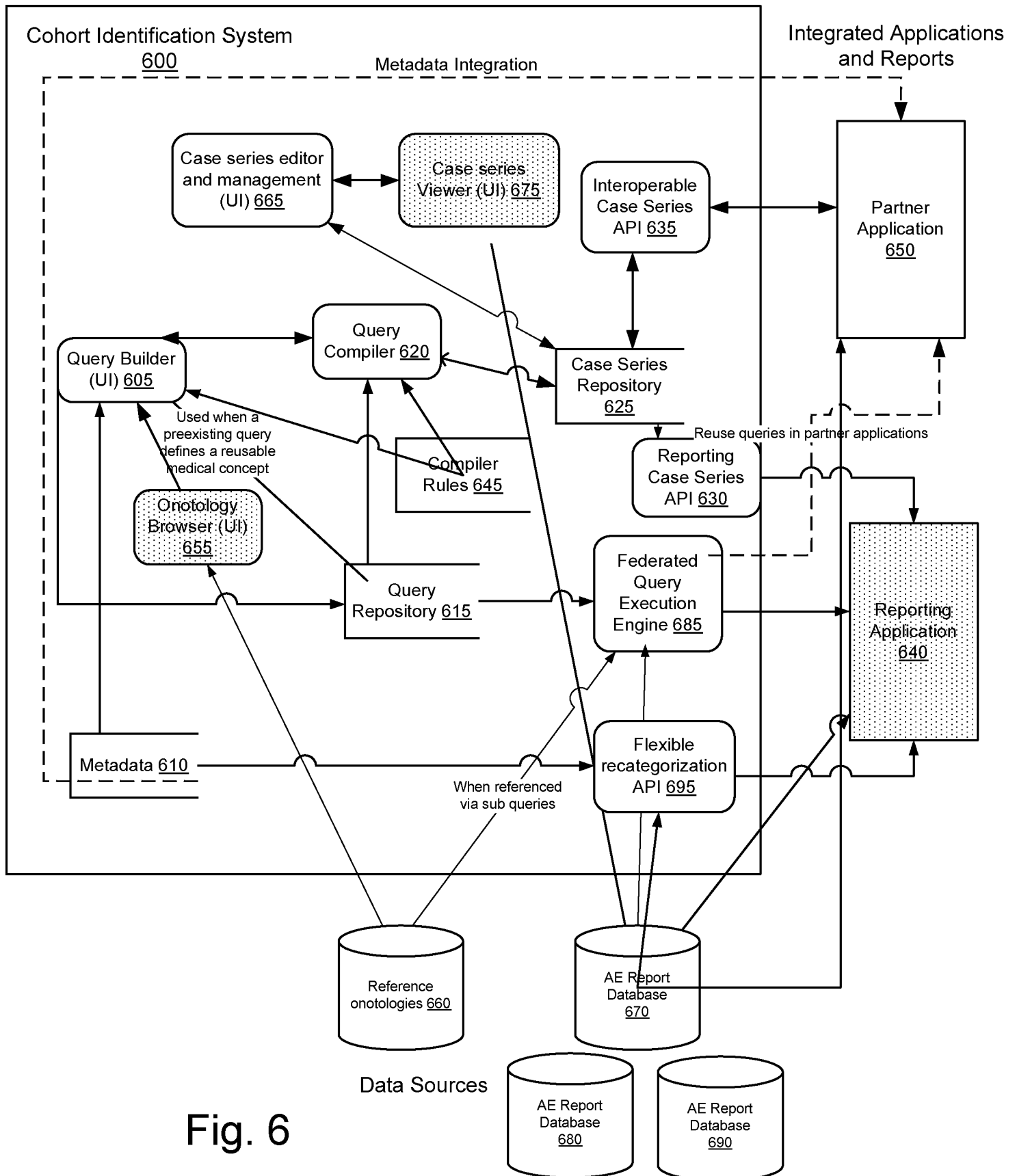


Fig. 5



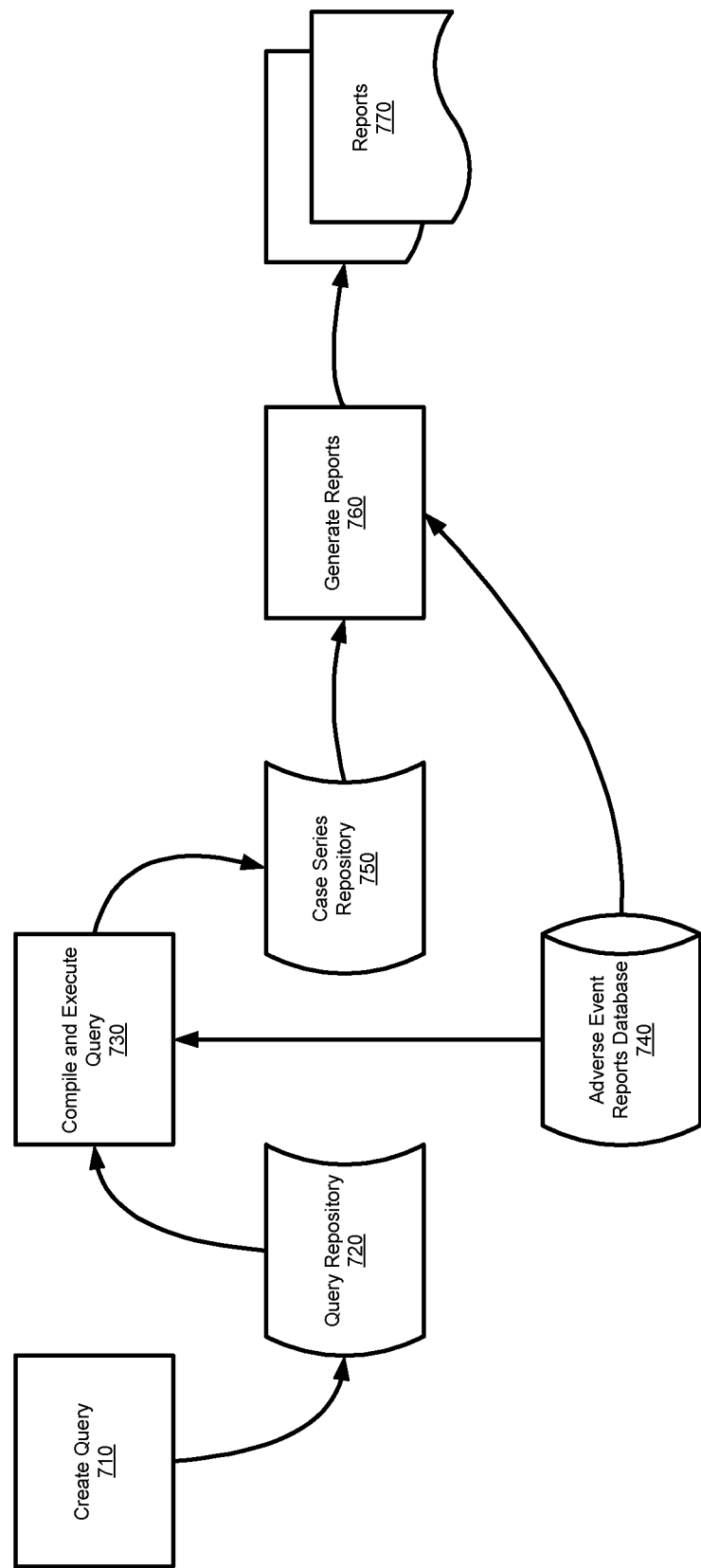


Fig. 7

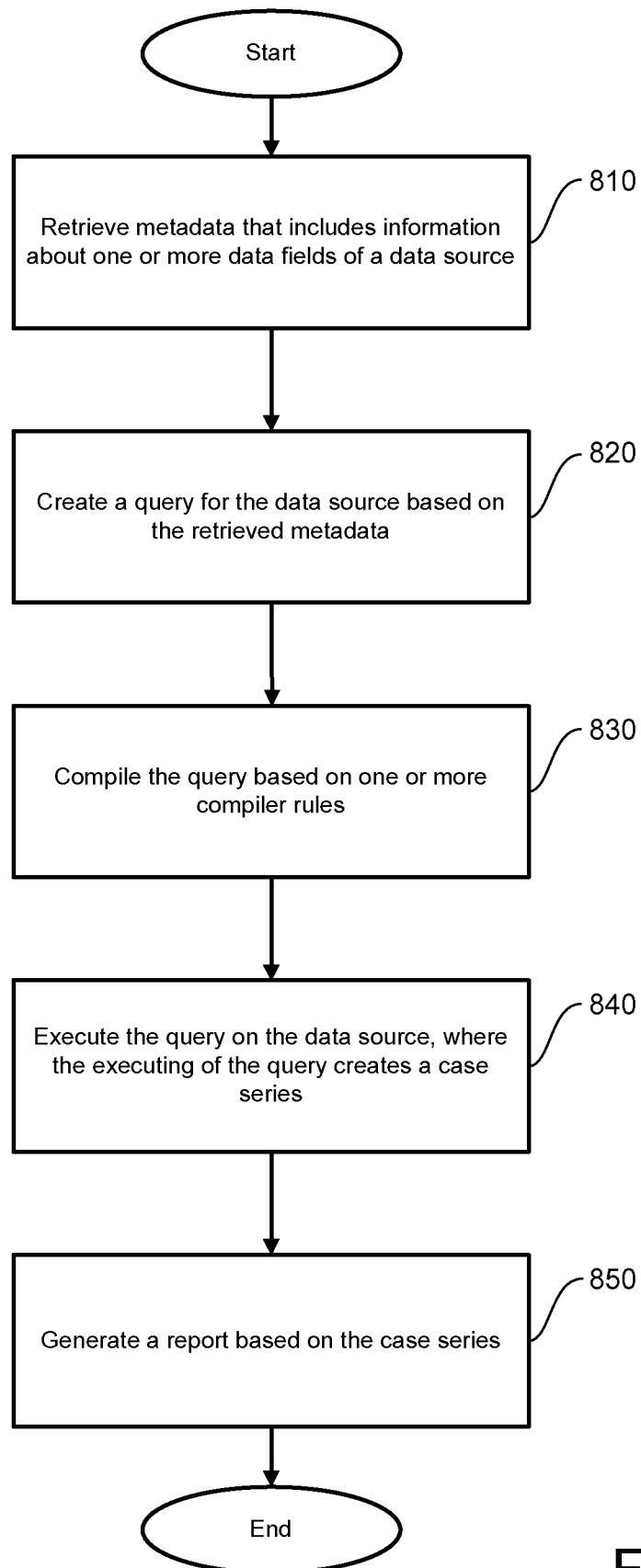


Fig. 8

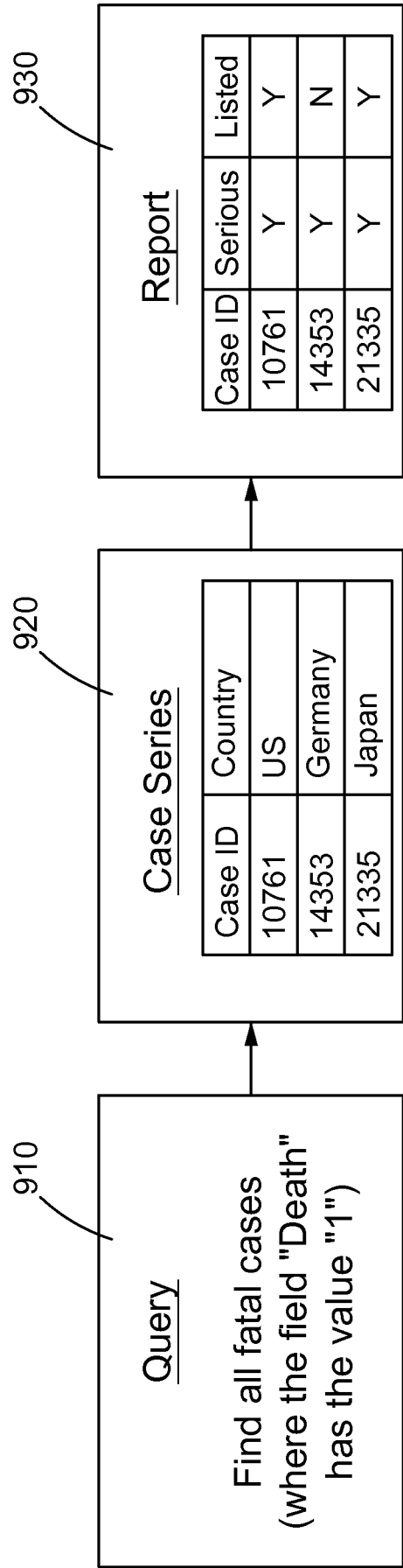


Fig.9

PATENT COOPERATION TREATY

PCT

DECLARATION OF NON-ESTABLISHMENT OF INTERNATIONAL SEARCH REPORT

(PCT Article 17(2)(a), Rules 13ter.1(c) and Rule 39)


Applicant's or agent's file reference T9049-19421 WO02	IMPORTANT DECLARATION	Date of mailing(<i>day/month/year</i>) 14 January 2014 (14-01-2014)
International application No. PCT/US2013/054836	International filing date(<i>day/month/year</i>) 14 August 2013 (14-08-2013)	(Earliest) Priority date(<i>day/month/year</i>) 31 October 2012 (31-10-2012)
International Patent Classification (IPC) or both national classification and IPC G06Q50/22		
Applicant ORACLE INTERNATIONAL CORPORATION		

This International Searching Authority hereby declares, according to Article 17(2)(a), that **no international search report will be established** on the international application for the reasons indicated below

1. ☐ The subject matter of the international application relates to:
 - a. ☐ scientific theories
 - b. ☐ mathematical theories
 - c. ☐ plant varieties
 - d. ☐ animal varieties
 - e. ☐ essentially biological processes for the production of plants and animals, other than microbiological processes and the products of such processes
 - f. ☐ schemes, rules or methods of doing business
 - g. ☐ schemes, rules or methods of performing purely mental acts
 - h. ☐ schemes, rules or methods of playing games
 - i. ☐ methods for treatment of the human body by surgery or therapy
 - j. ☐ methods for treatment of the animal body by surgery or therapy
 - k. ☐ diagnostic methods practised on the human or animal body
 - l. ☐ mere presentations of information
 - m. ☐ computer programs for which this International Searching Authority is not equipped to search prior art
2. ☒ The failure of the following parts of the international application to comply with prescribed requirements prevents a meaningful search from being carried out:

☒ the description
☐ the claims
☒ the drawings
3. ☐ A meaningful search could not be carried out without the sequence listing; the applicant did not, within the prescribed time limit:

☐ furnish a sequence listing on paper complying with the standard provided for in Annex C of the Administrative Instructions, and such listing was not available to the International Searching Authority in a form and manner acceptable to it.
☐ furnish a sequence listing in electronic form complying with the standard provided for in Annex C of the Administrative Instructions, and such listing was not available to the International Searching Authority in a form and manner acceptable to it.
☐ pay the required late furnishing fee for the furnishing of a sequence listing in response to an invitation under Rule 13ter.1(a) or (b).
4. Further comments:

Name and mailing address of the International Searching Authority  European Patent Office, P.B. 5818 Patentlaan 2 NL-2280 HV Rijswijk Tel. (+31-70) 340-2040 Fax: (+31-70) 340-3016	Authorized officer SOMMERMEYER, Katrin Tel: +49 (0)89 2399-7677
---	---

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 203

The claimed subject-matter, with due regard to the description and drawings in accordance with Rule 33.3 PCT, relates to processes comprised in the list of subject-matter and activities for which no search is required under Rule 39 PCT. The information technology employed as an enabler for carrying out said processes is conventional. Its use for carrying out non-technical processes forms part of common general knowledge and it was widely available to everyone at the date of filing of the present application. No documentary evidence is therefore considered necessary. (See Official Journal EP0 11/2007, pages 592ff and 594ff)

Specifically, the independent claims disclose a method, medium and system comprising the steps of:

- receiving a data item relating to an adverse event from a repository
- receiving a change to that data item
- storing the data item and the change in association with each other using an unknown data model
- allowing the stored data item and change to be retrieved by a software application via an API (Application Programming Interface).

These steps amount to an administrative procedure implemented by computing means. The computing means, a database (repository) used to store information in order that it can be retrieved subsequently by a software application, is notorious.

The applicant's attention is drawn to the fact that claims relating to inventions in respect of which no international search report has been established need not be the subject of an international preliminary examination (Rule 66.1(e) PCT). The applicant is advised that the EPO policy when acting as an International Preliminary Examining Authority is normally not to carry out a preliminary examination on matter which has not been searched. This is the case irrespective of whether or not the claims are amended following receipt of the search report or during any Chapter II procedure. If the application proceeds into the regional phase before the EPO, the applicant is reminded that a search may be carried out during examination before the EPO (see EPO Guidelines C-IV, 7.2), should the problems which led to the Article 17(2) declaration be overcome.