

KNOWLEDGE GRAPH CONSTRUCTION METHODS AND SYSTEMS

TECHNICAL FIELD

[0001] This specification relates to the field of computer technologies, and in particular, to
5 knowledge graph construction methods and systems.

BACKGROUND

[0002] Data flow scenarios such as fund flows between different fund accounts, goods
transport between different warehouses, and user travel and migration between different locations
often occur in daily life and production. With data flows, flow tracks and information are generated.

10 These data hide a large amount of valuable information, such as user preferences, behavioral habits,
and behavioral intentions. Recording these data can help analyze users' behaviors, make decisions,
and so on, thereby creating more convenience or value for life and production.

[0003] However, as time elapses, more and more data are generated. How to efficiently use
the massive data becomes a problem to be solved.

SUMMARY

15 [0004] One or more embodiments of this specification provide a knowledge graph
construction method. The method includes the following: determining an underlying graph based
on flow data, where the underlying graph includes entity nodes and an edge, and the edge includes
a flow edge that reflects data's flow relationship between the entity nodes; determining, based on
20 the underlying graph, one or more higher-layer graphs having an upper-lower hierarchical
relationship, where entity nodes of a lower-layer graph are subordinate to entity nodes of an upper-
layer graph; and storing the underlying graph and the one or more higher-layer graphs for query.

[0005] One or more embodiments of this specification provide a knowledge graph
construction system. The system includes the following: an underlying-graph determining module,

configured to determine an underlying graph based on flow data, where the underlying graph includes entity nodes and an edge, and the edge includes a flow edge that reflects data's flow relationship between the entity nodes; a higher-layer-graph determining module, configured to determine, based on the underlying graph, one or more higher-layer graphs having an upper-lower hierarchical relationship, where entity nodes of a lower-layer graph are subordinate to entity nodes of an upper-layer graph; and a graph storage module, configured to store the underlying graph and the one or more higher-layer graphs for query.

[0006] One or more embodiments of this specification further provide a data query method. The method includes the following: obtaining a query request, where the query request includes a query graph; and obtaining, based on the query graph and from the higher-layer graph obtained by using the knowledge graph construction method described above, sub-graph data used to generate a query result.

[0007] One or more embodiments of this specification further provide a data query system. The system includes the following: a query request acquisition module, configured to obtain a query request, where the query request includes a query graph; and a query result acquisition module, configured to obtain, based on the query graph and from the higher-layer graph obtained by using the knowledge graph construction method described above, sub-graph data used to generate a query result.

[0008] One or more embodiments of this specification provide a knowledge graph construction apparatus, including a processor, where the processor is configured to perform the knowledge graph construction method.

[0009] One or more embodiments of this specification provide a computer-readable storage medium, where the storage medium stores a computer instruction, and after a computer reads the computer instruction in the storage medium, the computer performs the knowledge graph construction method.

BRIEF DESCRIPTION OF DRAWINGS

[0010] This specification will be further illustrated by way of example embodiments. These example embodiments will be described in detail with reference to the accompanying drawings. These embodiments are not limitative. Identical reference numerals in these embodiments

represent the same structures, where:

[0011] FIG. 1 is an example schematic diagram illustrating an underlying graph involved in some embodiments of this specification;

[0012] FIG. 2 is an example schematic diagram illustrating a knowledge graph construction method according to some embodiments of this specification;

[0013] FIG. 3 is an example schematic diagram illustrating determining of higher-layer graph according to some embodiments of this specification;

[0014] FIG. 4 is an example schematic diagram illustrating determining of an edge feature of a flow edge of an upper-layer graph according to some embodiments of this specification;

[0015] FIG. 5 is an example schematic diagram illustrating determining of a control feature of an upper-layer controlled node according to some embodiments of this specification;

[0016] FIG. 6 is an example module diagram illustrating a knowledge graph construction system according to some embodiments of this specification; and

[0017] FIG. 7 is an example module diagram illustrating a data query system according to some other embodiment of this specification.

DESCRIPTION OF EMBODIMENTS

[0018] To describe the technical solutions in the embodiments of this specification more clearly, the following briefly describes the accompanying drawings needed for describing the embodiments. Clearly, the accompanying drawings in the following description show merely some examples or embodiments of this specification, and those of ordinary skill in the art can further apply this specification to other similar scenarios based on these accompanying drawings without creative efforts. Unless clearly identified from a language environment or otherwise stated, identical reference numerals in the drawings represent the same structure or operation.

[0019] It should be understood that "system", "apparatus", "unit", and/or "module" used in this specification are/is an approach used to distinguish between different components, elements, parts, portions, or assemblies of different levels. However, if other terms can achieve the same purpose, these terms can be replaced by other expressions.

[0020] As illustrated in this specification and the claims, the words "an", "one", "one type", and/or "the" do not specifically refer to singular forms and can also include plural forms unless

otherwise explicitly stated in the context. Generally speaking, the terms "include" and "comprise" only indicate inclusion of those steps and elements that have been explicitly identified, and these steps and elements do not constitute an exclusive listing, and a method or device may also include other steps or elements.

5 **[0021]** A flowchart is used in this specification to describe operations performed by a system according to some embodiments of this specification. It should be understood that former or latter operations are not necessarily performed exactly according to a sequence. Instead, the steps can be processed in a reverse sequence or simultaneously. In addition, other operations can also be added to these processes, or a step or several operations can also be removed from these processes.

10 **[0022]** Data flow scenarios such as fund flows between different fund accounts, goods transport between different warehouses, and user travel and migration between different locations often occur in daily life and production. These scenarios generate a large amount of data flow data, and the data contain rich and valuable information.

15 **[0023]** Using funds as an example, funds can be transferred between a user and an enterprise, between users, and between enterprises in a certain way. Fund transfer can also be referred to as fund flow. By analyzing fund flow data, an intention of fund transfer by a user or an enterprise can be known, and illegal acts such fund collection by improper approaches or use of funds for abnormal channels can be discovered in time. Therefore, fund flow information is analyzed. For example, where does each fund in a fund flow process come from (recharge, transfer, etc.), where
20 has the fund passed through, how long has the fund been flowing, and where does the fund finally flows to, so as to know a destination the fund, whether use channel of the fund is illegal, a source of funds in a certain fund account, and whether the source of these funds is normal.

25 **[0024]** As time elapses, a quantity of flows is increasing, flow information gradually increases, and a size of fund data is also growing. Existing analysis on a source and a destination of a fund is performed based on a fund business table, and one transaction is stored as one record. The fund business table records transaction feature information such as a receipt/payment, a scenario, a time, and an amount of a transaction. A storage method of the fund business table is essentially a relationship-based data storage method. Transaction data are characterized by a data flow and fund propagation. When a fund flow needs to be analyzed for a business, multiple join operations often
30 need to be performed on the same table/multiple tables. Each time fund flow analysis is added, a transaction table needs to be associated once. Such practice not only multiplies calculation

resources and causes the analysis to be time-consuming, but also greatly increases maintenance costs of analysis code.

[0025] It should be noted that, the above-mentioned fund flow example is merely for the purpose of illustration, and is not intended to limit application scenarios of the technical solutions disclosed in this specification. For example, the above-mentioned fund flow example can be further used to analyze call fees, traffic, goods transport, a traveler flow, a vehicle flow, etc.

[0026] FIG. 1 is an example schematic diagram illustrating an underlying graph involved in some embodiments of this specification.

[0027] In some embodiments, an underlying base graph can be constructed based on flow data to represent a data flow, and the underlying base graph is referred to as an underlying graph. A graph is a semantic network data structure that reveals a relationship between entities (or objects). The graph can include nodes and an edge. The nodes represent entities, and the nodes can have multiple types, referred to as node types, which are used to indicate various entities. The entities can indicate specific or abstract objects in the real world, such as people, accounts, sub-accounts subordinate to accounts, place names, concepts, drugs, etc. The edge represents a relationship, and the edge can also have multiple types, referred to as edge types, which are used to indicate various relationships. The relationship can be used to express an association between different entities, for example, Zhang San and Li Si are in a "friends relationship", or in a "transaction relationship" in which Zhang San transfers 50 RMB to Li Si. An edge of the underlying graph can include a flow edge that reflects data's flow relationship between nodes.

[0028] The flow data can be an abstract reference to relevant information about a person or object that flows between different nodes. For example, the flow data can be data such as funds, call fees, traffic, people, goods, vehicles, etc.

[0029] As shown in FIG. 1, an underlying graph 100 can include entity nodes 110 and a flow edge 120.

[0030] The entity nodes 110 can be configured to represent various entities. In some embodiments, the entity nodes 110 can be configured to represent correspondences between various entities, such as accounts (for example, bank accounts, application accounts, etc. that can store funds), locations, warehouses, buildings, people, companies, etc. In some embodiments, one entity node can represent a sub-account or product (for example, YUE BAO, a balance, ANT CREDIT PAY, ANT CASH NOW, etc. of ALIPAY) subordinate to one account. In some

embodiments, one or more entity nodes can correspond to the same account, or can correspond to different accounts. For example, 130 can represent a first account, 140 can represent a second account, and 150 can represent a third account. Each account has respective corresponding entity nodes.

5 **[0031]** The flow edge 120 can be configured to represent data's flow between the entity nodes 110. The flow edge 120 can be a directed edge, which points from one entity node to another. In some embodiments, the flow edge 120 can further represent data's flow scenario between the entity nodes 110, for example, a way of data flow. For example, the data are fund data. When the data are funds, call fees, traffic, etc., the data can flow from node 1 to node 3. Node 1 can represent a balance of a user's ALIPAY account. Node 2 represents YU'E BAO of the user's ALIPAY. Node 3 represents another user's bank card. Then, node 1 → node 2 → node 3 can represent that a fund from the user's balance flows to the user's YU'E BAO account by way of YU'E BAO purchase, and then flows to the another user's bank card by way of transfer. Balance → YU'E BAO → bank card indicates a path through which the fund data flow. In some embodiments, the flow edge can further include scenario information such as a direction and transfer of fund data flow, recharge, consumption, etc. For example, a flow edge of node 1 → node 2 can indicate that an amount of 10 RMB flows from the balance to YU'E BAO by way of YU'E BAO purchase on January 1, 2021, and a flow edge of node 2 → node 3 can indicate that 10 RMB in YU'E BAO is transferred to the bank card by way of transfer on January 2, 2021. Node 1 → node 2 → node 3 indicates that a fund of 10 RMB of node 1 passes through node 2 and node 3 via a corresponding flow scenario (such as transfer and recharge), and finally flows to node 3. For another example, an entity node can represent a location, and a flow edge can represent a moving direction and a moving mode of a person and a vehicle. For another example, user travel and migration are used as an example. Node 1 can represent Beijing, node 2 can represent Shanghai, and node 3 can represent Guangzhou. In this case, node 1 → node 2 → node 3 can indicate that a certain user passes through Shanghai in certain ways (for example, walking, riding a bike, taking a taxi, taking public transportation, etc.) from Beijing, and reaches Guangzhou at a specific time.

30 **[0032]** A flow between entity nodes can be an internal flow in a same account. For example, the above-mentioned fund data flow represented by node 1 → node 2 → node 3 can represent an internal flow in the first account 130. Or, it can be a flow between different accounts. For example, node 1 → node 5 → node 6 in FIG. 1 can represent a flow between the first account 130 and the

third account 150.

[0033] FIG. 2 is an example schematic diagram illustrating a knowledge graph construction method according to some embodiments of this specification.

[0034] In some embodiments, a processing device can establish an underlying graph based on flow data. The flow data can include structured flow data, for example, transaction relationship data stored in a form of a data table, or unstructured flow data, for example, flow data recorded in a form of text or a picture.

[0035] For example, an underlying graph is created based on fund transaction flow data. The processing device can extract, from structured fund flow data, a field (for example, ANT CREDIT PAY, ANT CASH NOW, or YU'E BAO) corresponding to a sub-account for fund storage as an entity node for creating the underlying graph. An attribute value (also referred to as a node instance) of the entity node is a specific value corresponding to the field of the sub-account. A field (for example, transfer, recharge, or expense) corresponding to a transaction relationship in the fund flow data is used as an edge for creating the base graph, and a specific value of the field corresponding to the transaction relationship can be used as relationship data (also referred to as an edge instance) of the edge. Finally, the underlying graph is established based on the determined node and edge. In addition, a node feature of the underlying graph can be determined based on the node instance, and an edge feature of the underlying graph can be determined based on the edge instance.

[0036] The processing device can process unstructured flow data in various ways, and convert the unstructured flow data into structured data. For example, the processing device can extract information related to a fund transaction from text, and determine an underlying graph based on the structured flow data.

[0037] In the embodiments, recorded relationship-type flow data are converted into a graph structure, and capabilities such as analysis, mining, and algorithm inference of a graph can be used to satisfy application requirements of different scenarios so as to improve analysis efficiency and reduce resource overheads and time consumption of analysis and calculation.

[0038] In some embodiments, to better satisfy application requirements of different scenarios, the processing device can determine, based on the underlying graph, one or more higher-layer graphs having an upper-lower hierarchical relationship.

[0039] A higher-layer graph can be a graph constructed on the basis of the underlying graph.

In some embodiments, any constructed graph other than the underlying graph can be referred to as a higher-layer graph. The higher-layer graph includes entity nodes and an edge. There is a correspondence between the entity nodes and the edge of the higher-layer graph and the entity nodes and the edge of the underlying graph. For example, when the flow data are fund data, the entity nodes of the higher-layer graph correspond to accounts or users to which sub-accounts represented by the entity nodes of the underlying graph are subordinate, and the edge of the higher-layer graph corresponds to one or more flow edges of the underlying graph.

[0040] The upper-lower hierarchical relationship refers to a subordinate relationship between graphs, and entity nodes of a lower-layer graph are subordinate to entity nodes of an upper-layer graph. A concept of the upper-lower hierarchical relationship is a relative concept, which refers to a subordinate relationship between any two graphs. For example, any higher-layer graph is an upper-layer graph of the underlying graph. An upper-lower hierarchical relationship between multiple higher-layer graphs can be determined based on a subordinate relationship between entity nodes. Entity nodes of a lower-layer graph are subordinate to entity nodes of an upper-layer graph.

[0041] For example, a dashed arrow in FIG. 2 represents a subordinate relationship between a node of a lower-layer graph and a node of an upper-layer graph. In comparison between an underlying graph 202 and a higher-layer graph 204, the underlying graph 202 is a lower-layer graph, and the higher-layer graph 204 is an upper-layer graph. In comparison between the underlying graph 202 and a higher-layer graph 206, the underlying graph 202 is a lower-layer graph, and the higher-layer graph 206 is an upper-layer graph. For another example, in comparison between the higher-layer graph 204 and the higher-layer graph 206, because nodes of the higher-layer graph 204 are subordinate to nodes of the higher-layer graph 206, the higher-layer graph 204 is a lower-layer graph, and the higher-layer graph 206 is an upper-layer graph. In some embodiments, the upper-lower hierarchical relationship can also be a subordinate relationship between two non-adjacent graphs. For example, if one higher-layer graph is further constructed on the basis of the higher-layer graph 206, in comparison between the higher-layer graph 204 and this higher-layer graph, this higher-layer graph is an upper-layer graph, and the higher-layer graph 204 is a lower-layer graph.

[0042] In some embodiments, the processing device can determine a higher-layer graph based on the entity nodes and the edge of the underlying graph and instance data corresponding to the entity nodes and the edge. For example, statistics of the nodes and the flow edge of the underlying

graph as well as relationship data (for example, edge instance data) corresponding to each flow edge can be collected by way of traversal. In addition, entity nodes of a higher-layer graph can be determined based on the entity nodes of the underlying graph, and a flow edge of the higher-layer graph can be determined based on relationship data corresponding to each flow edge. For more
5 embodiments of constructing a higher-layer graph, reference can be made to related descriptions of FIG. 3.

[0043] In some embodiments, in addition to the entity nodes, a node of the underlying graph can further include another node, for example, used to indicate a type of a sub-account corresponding to an entity node of the underlying graph, for example, whether the sub-account is
10 a product sub-account of ALIPAY or a product sub-account of a bank card. Similarly, in addition to the entity nodes, a node of a higher-layer graph can also include another node.

[0044] In some embodiments, types of the entity nodes and the edge of the underlying graph can correspond to a type of the flow data. For example, when the flow data are fund data, the entity nodes of the underlying graph can correspond to sub-accounts for fund storage, and the edge of
15 the underlying graph can reflect a flow of the fund data between the sub-accounts, for example, from one sub-account to another. For another example, when the flow data are logistics data, the entity nodes of the underlying graph can correspond to sub-warehouses for material storage, and the edge of the underlying graph can reflect a transfer of materials between the sub-accounts. Entity nodes of a higher-layer graph correspond to accounts or users to which the sub-accounts
20 corresponding to the entity nodes of the underlying graph are subordinate. For example, a sub-account is subordinate to an account, or a sub-account is subordinate to a user. The user can be a natural person or an enterprise.

[0045] In some implementations of this specification, by abstracting and modeling flow data generated by a business, structured flow data are converted into a knowledge graph of a graph
25 structure, and one or more higher-layer graphs are established hierarchically. Higher-layer graphs of different layers are applied to different requirement scenarios to satisfy requirements from a perspective of differentiated network transactions.

[0046] In some embodiments, after determining one or more higher-layer graphs, the processing device can store the underlying graph and the one or more higher-layer graphs for query
30 by users. For example, a user can select a graph of a certain layer according to a business requirement, for example, the higher-layer graph 204 or the higher-layer graph 206 to obtain

desired sub-graph data by query, and perform business analysis based on the sub-graph data in order to help the user quickly learn more information and assist the user in making a business decision.

[0047] In some embodiments, the storage can be storage to a storage device, a database, or a specified address.

[0048] In some embodiments, the processing device can perform a query in a way described in the following embodiments.

[0049] The processing device can obtain a query request. The query request can include a query graph. The query graph defines, from a perspective of a knowledge graph schema, nodes, an edge, and a mutual connection relationship from which instance data need to be looked up. The query graph can correspond to a query request of a user, and can reflect a query condition input by the user. A core issue of lookup using a query graph is to determine whether graph data (such as a data graph) contain a sub-graph that satisfies the nodes, the edge, and the connection relationship thereof described in the query graph. Therefore, this process is also referred to as sub-graph matching or a graph query. In some embodiments, the processing device can obtain the query request by receiving a user input. The processing device can obtain, based on the query graph, sub-graph data from the stored higher-layer graph so as to generate a query result. The query result can include sub-graph data obtained by querying a graph based on the query graph and/or information (such as statistics and features) determined based on the sub-graph data. This information can help the user more quickly understand content of the sub-graph data obtained by query, and improve the user's efficiency of business analysis.

[0050] For example, in fund risk assessment, when it is learned that a gambler participates in a gambling behavior, a source of funds in the gambler's account needs to be tracked. In this case, according to a user's need, for example, a need to know a specific person or account whose funds flow to the gambler's account, sub-graph data can be obtained from a higher-layer graph for analysis. If the user needs to know a specific account (for example, needs to be accurate to a sub-account level) whose funds flow to the gambler's account, sub-graph data can be obtained from a low-layer graph for analysis. Such practice satisfies different business requirements of the user. Moreover, performing a query by way of a graph can enable a wider query range and higher efficiency.

[0051] FIG. 3 is an example schematic diagram illustrating determining of higher-layer graph

according to some embodiments of this specification. Referring to FIG. 3, 310 represents an entity node of a lower-layer graph, and node 330 represents an entity node, of an upper-layer graph, that corresponds to the entity node 310 of the lower-layer graph.

[0052] In some embodiments, the processing device can determine entity nodes of the upper-layer graph based on labels corresponding to entity nodes of the lower-layer graph.

[0053] A label corresponding to an entity node can refer to an identifier used to distinguish each entity node. The label can be determined based on an attribute value of the entity node. For example, an entity node corresponds to an account. An attribute value of the account can include an account ID and an account type. A user can select the account ID to distinguish this entity node from another, or can select the account type to distinguish this entity node from another. The user can perform selection according to an actual need.

[0054] The label corresponding to an entity node can be used to distinguish a subordinate relationship between entity nodes. For example, entity nodes of a lower-layer graph can be sub-accounts of application software used to store fund data, and entity nodes of an upper-layer graph can be accounts of the application software. In this case, the entity nodes of the lower-layer graph can have identifiers to indicate that the sub-accounts corresponding to the entity nodes of the lower-layer graph are subordinate to the same account corresponding to an entity node of the upper-layer graph. For example, payment software is used as an example. The payment software is usually provided with multiple products, and each product can correspond to one sub-account, for example, ANT CREDIT PAY or YU'E BAO. These sub-accounts can store fund data. In addition, these sub-accounts can be subordinate to an account of the payment software, for example, 2088.... An account ID corresponding to an entity node of an upper-layer graph can be used as a label corresponding to an entity node of a lower-layer graph. For another example, one or more accounts corresponding to a node of an upper-layer graph can be subordinate to an entity node of an upper-layer graph of a higher layer. For example, an account corresponding to application software of the node of the upper-layer graph can be subordinate to a certain natural person, and an identifier of the natural person can be a certificate number of the natural person. In this case, the certificate number can be used as a label corresponding to a graph node at an account level.

[0055] In some embodiments, the processing device can determine an entity node of an upper-layer graph based on whether labels corresponding to one or more entity nodes of a lower-layer

graph are the same. For example, the processing device can correspond one or more nodes with the same label to an entity node of an upper-layer graph.

[0056] In some embodiments, the processing device can determine an upper-layer flow edge based on flow endpoints of a lower-layer flow edge and entity nodes, of the upper-layer graph, to which the flow endpoints are subordinate.

[0057] Flow endpoints can refer to a start node and an end node of a flow edge. For example, 320 can represent one flow edge, and two nodes of the edge are flow endpoints.

[0058] A lower-layer flow edge is a flow edge of a lower-layer graph. For example, an edge between flow endpoints (nodes) of the lower-layer graph shown in FIG. 3 is a lower-layer flow edge, for example, the lower-layer flow edge 320. An upper-layer flow edge is a flow edge of an upper-layer graph. For example, an edge between flow endpoints (nodes) of the upper-layer graph shown in FIG. 3 is an upper-layer flow edge, for example, an upper-layer flow edge 340.

[0059] In some embodiments, for one lower-layer flow edge, if two flow endpoints of the lower-layer flow edge are subordinate to different nodes of an upper-layer graph, a flow edge is determined between the nodes of the upper-layer graph to which the two flow endpoints of the lower-layer flow edge are subordinate. For example, two flow endpoints of the lower-layer flow edge 320 are respectively subordinate to a node corresponding to account B of the upper-layer graph and a node corresponding to account C of the upper-layer graph. In this case, it can be determined that there is an upper-layer flow edge between account B and account C. For another example, if nodes, of the upper-layer graph, to which two flow endpoints of a lower-layer flow edge 350 are subordinate to are both a node corresponding to account C, no upper-layer flow edge is determined on the node of account C.

[0060] For more description of determining a flow edge of an upper-layer graph, reference can be made to the following detailed description of FIG. 4.

[0061] FIG. 4 is an example schematic diagram illustrating determining of an edge feature of a flow edge of an upper-layer graph according to some embodiments of this specification.

[0062] In some embodiments, the processing device can determine an edge feature of an upper-layer flow edge based on a lower-layer flow edge and a subordinate relationship between a lower-layer graph and an upper-layer graph. Endpoints of the lower-layer flow edge include a first lower-layer endpoint and a second lower-layer endpoint, endpoints of the upper-layer flow edge include a first upper-layer endpoint and a second upper-layer endpoint, the first lower-layer endpoint is

subordinate to the first upper-layer endpoint, and the second lower-layer endpoint is subordinate to the second upper-layer endpoint. In some embodiments, the first lower-layer endpoint represents a start point of the lower-layer flow edge, the second lower-layer endpoint represents an end point of the lower-layer flow edge, the first upper-layer endpoint represents a start point of the upper-layer flow edge, and the second upper-layer endpoint represents an end point of the upper-layer flow edge.

[0063] For example, 410 and 420 in FIG. 4 respectively represent lower-layer flow edges. In endpoints of the lower-layer flow edge 410 and the lower-layer flow edge 420, first lower-layer endpoints and second lower-layer endpoints are respectively subordinate to different endpoints of an upper-layer flow edge. For example, the first lower-layer endpoints are subordinate to a first upper-layer endpoint of an upper-layer flow edge 430, and the second lower-layer endpoints are subordinate to a second upper-layer endpoint of the upper-layer flow edge 430. Therefore, an edge feature of the upper-layer flow edge 430 can be determined based on edge features of the lower-layer flow edge 410 and the lower-layer flow edge 420.

[0064] In some embodiments, the processing device can extract and summarize an edge feature of one or more lower-layer flow edges to determine an edge feature of an upper-layer flow edge. For example, assuming that the edge feature corresponding to the lower-layer flow edge 410 is an expense of 30 RMB and a time of 19: 00, and the edge feature corresponding to the lower-layer flow edge 420 is a transfer of 50 RMB and a time of 19: 05, the edge feature of the upper-layer flow edge 430 corresponding to the lower-layer flow edge 410 and the lower-layer flow edge 420 can include the edge features of the lower-layer flow edge 410 and the lower-layer flow edge 420. For example, the edge features of the lower-layer flow edges 410 and 420 can be directly used as the edge feature of the upper-layer flow edge, or can be extracted and summarized to obtain the edge feature corresponding to the upper-layer flow edge 430. For example, by using the above-mentioned example, after feature extraction and summarization are performed on the edge features of the lower-layer flow edges, the edge feature corresponding to the upper-layer flow edge can be obtained as follows: a total amount of 80 RMB, two times in total, a scenario feature of 37.5% for expense and 62.5% for transfer, and a time feature of 18: 00–19: 00 for 37.5% and 19: 00–20: 00 for 62.5%.

[0065] In the embodiments, detailed information of the flow data is precipitated into the upper-layer graph by performing extraction and summarization based on the lower-layer graph. For

example, extracting transaction relationships corresponding to the lower-layer flow edge 410 and the lower-layer flow edge 420 to summarize key transaction information not only enables more convenient business analysis for users, but also reduces copy of massive transaction data and storage of duplicate data.

5 **[0066]** FIG. 5 is an example schematic diagram illustrating determining of a control feature of an upper-layer controlled node according to some embodiments of this specification.

[0067] In some embodiments, the processing device can determine, based on the lower-layer graph, a lower-layer controlled node that satisfies a control condition.

[0068] A controlled node can refer to a node labeled in a graph. This node can be a node
10 representing a high-quality user, or can be a node representing a risk user. For example, a risk user is used as an example. If a certain user uses his/her sub-account or account of certain application software to perform a behavior such as money laundering or illegal fundraising, when it is determined that the user is involved in the illegal behavior, a node on which the sub-account or the account corresponding to the user is located is labeled as a controlled node (a label can be manually
15 added). A node whose relationship with this node satisfies a control condition is looked up from the graph. For example, the control condition can be as follows: There is a direct connection relationship with the controlled node; or there is connection relationship within N degrees with the controlled node, where N can be 2, 3, 4, 5, etc.; or a flow relationship with the controlled node forms a closed loop, for example, flow data flow out from the controlled node and finally return
20 to the controlled node. A connection relationship within N degrees indicates that there may be a close relationship with the controlled node. Forming a closed loop may involve money laundering and the like by way closed-loop flow, and all nodes on the closed-loop path may participate in money laundering. Labeling these nodes as controlled nodes can help users quickly learn, by using the controlled nodes, that these nodes need special attention.

25 **[0069]** In some embodiments, the lower-layer controlled node can be represented in multiple ways. For example, the controlled node can be labeled by adding a label or a node feature, or can be labeled by using a color label or a graphical label in a graphical representation of the lower-layer controlled node. This specification sets no limitation thereto. As an example only, in FIG. 5, a controlled node is represented by a square node, while a circle represents a non-controlled node.

30 **[0070]** In some embodiments, the processing device can determine, based on a controlled node in the lower-layer graph and a corresponding control condition, a lower-layer controlled node that

satisfies the control condition.

[0071] In some embodiments, the processing device can determine that an entity node, of the upper-layer graph, to which the lower-layer controlled node is subordinate is an upper-layer controlled node. For example, if a controlled node 510 is subordinate to a node 520 of an upper-layer graph, the node 520 can be determined as an upper-layer controlled node.

[0072] In some embodiments, the processing device can determine an upper-layer controlled node based on a correspondence between an entity node of the lower-layer graph and an entity node of the upper-layer graph.

[0073] It should be noted that, a node can be determined as an upper-layer controlled node when one or more nodes of the lower-layer graph that correspond to this node are controlled nodes. For example, even if illegal funds are transferred only by using a balance product in a certain account, the account corresponding to the balance product can still be labeled.

[0074] In some embodiments, the processing device can determine a control feature of the upper-layer controlled node based on the control condition.

[0075] The control feature can be a node feature added to the controlled node of the upper-layer graph based on the control condition. For example, when the control condition can be that a relationship with the controlled node 510 satisfies a neighbor relationship within 3 degrees, a corresponding control feature can be being a 1-degree neighbor with the controlled node 510. For another example, when the control condition is that a flow relationship with the controlled node forms a closed-loop path, and can even be that there is a flow relationship with a closed-loop node (that is, there is an edge with the closed-loop node), the control feature of the upper-layer controlled node can be as follows: there is a closed-loop flow relationship with the controlled node 510 or there is a flow relationship with the controlled node 510.

[0076] In some embodiments, the control feature of the upper-layer controlled node corresponding to the lower-layer controlled node can further be determined based on the lower-layer controlled node. For example, if a user labels the controlled node 510 (which is involved in gambling as a banker) of the lower-layer graph, an entity node, of the upper-layer graph, corresponding to the controlled node 510 is accordingly the controlled node 520. In this case, a node feature of the controlled node 520 can be "gambling banker", and a node feature of a controlled node that satisfies the control condition can be "gambler".

[0077] In some embodiments, a user can obtain sub-graph information from a data flow graph

based on a query condition (or referred to as a query request). For example, if a path matching condition in a query request includes "node 1 → node 2", it indicates that all flow paths including the path from node 1 to node 2 need to be obtained from the data flow graph. By performing a traversal query in the entire graph, all sub-graph data that satisfy the above-mentioned path matching condition can be obtained, and the obtained sub-graph data can be used for analysis of a fund flow direction and a fund source, a fund security analysis (for example, whether money laundering is involved), or the like.

[0078] It should be noted that the descriptions above are merely used as examples and illustrations, and are not intended to limit the scope of this specification. Those skilled in the art can make various amendments and changes to the embodiments under the guidance of this specification. However, these amendments and changes still fall within the scope of this specification.

[0079] FIG. 6 is an example module diagram illustrating a knowledge graph construction system according to some embodiments of this specification. As shown in FIG. 6, a system 600 can include an underlying-graph determining module 610, a higher-layer-graph determining module 620, and a graph storage module 630.

[0080] The underlying-graph determining module 610 can be configured to determine an underlying graph based on flow data, where the underlying graph includes entity nodes and an edge, and the edge includes a flow edge that reflects data's flow relationship between the entity nodes.

[0081] The higher-layer-graph determining module 620 can be configured to determine, based on the underlying graph, one or more higher-layer graphs having an upper-lower hierarchical relationship, where entity nodes of a lower-layer graph are subordinate to entity nodes of an upper-layer graph.

[0082] The graph storage module 630 can be configured to store the underlying graph and the one or more higher-layer graphs for query.

[0083] FIG. 7 is an example module diagram illustrating a data query system according to some other embodiment of this specification. As shown in FIG. 7, a system 700 can include a query request acquisition module 710 and a query result acquisition module 720.

[0084] The query request acquisition module 710 can be configured to obtain a query request, where the query request includes a query graph.

[0085] The query result acquisition module 720 can be configured to obtain, based on the query graph and from the higher-layer graph obtained by using the knowledge graph construction method according to some other embodiment of this specification, sub-graph data used to generate a query result.

5 **[0086]** For a specific description of the modules of the system illustrated above, reference can be made to the flowchart part of this specification, for example, related descriptions of FIG. 2 to FIG. 5.

[0087] It should be understood that the system and its modules shown in FIG. 6 can be implemented in various ways. For example, in some embodiments, the system and its modules can be implemented by software, hardware, or a combination of software and hardware. A hardware portion can be implemented by using dedicated logic. A software portion can be stored in a memory and executed by an appropriate instruction execution system, such as a microprocessor or specially designed hardware. Those skilled in the art can understand that the above-mentioned methods and systems can be implemented by using computer-executable instructions and/or by being contained in processor-controlled code, which is provided on carrier media such as a magnetic disk, a CD, or a DVD-ROM, programmable memories such as a read-only memory (firmware), or data carriers such as an optical or electronic signal carrier. The system and its modules of this specification can be implemented by very large scale integrated circuits or gate arrays, semiconductors such as logic chips and transistors, or hardware circuits of programmable hardware devices such as field programmable gate arrays and programmable logic devices, or can be implemented by, for example, software executed by various processors, or can be implemented by a combination of the above-mentioned hardware circuits and software (for example, firmware).

[0088] It should be noted that the above-mentioned description of the knowledge graph construction system and its modules is for ease of description only, and cannot limit this specification to the scope of the embodiments described. It can be understood that, after understanding a principle of the system, those skilled in the art may randomly combine the modules or construct a connection between a subsystem and another module without departing from the principle. For example, in some embodiments, the underlying-graph determining module 610, the higher-layer-graph determining module 620, and the graph storage module 630. For example, the modules can share one storage module, or the modules can have their own storage modules. Such variations all fall within the protection scope of this specification.

[0089] It should be noted that different embodiments may generate different beneficial effects. In different embodiments, a possible beneficial effect can be any one or a combination of the above-mentioned effects, or can be any other beneficial effect that can possibly be obtained.

[0090] The basic concepts have been described above. Clearly, for those skilled in the art, the above detailed disclosure is only an example, and does not constitute a limitation to this specification. Although not explicitly stated here, those skilled in the art may make various modifications, improvements, and amendments to this specification. Such modifications, improvements, and amendments are intended to be suggested in this specification, so such modifications, improvements, and amendments still belong to the spirit and scope of the example embodiments of this specification.

[0091] Moreover, certain terminology has been used to describe embodiments of this specification. For example, "one embodiment", "an embodiment", and/or "some embodiments" refer to a certain feature, structure, or characteristic related to at least one embodiment of this specification. Therefore, it should be emphasized and noted that two or more references to "an embodiment", "one embodiment", or "an alternative embodiment" in various places in this specification are not necessarily referring to the same embodiment. In addition, certain features, structures, or characteristics of one or more embodiments of this specification can be properly combined.

[0092] In addition, those skilled in the art may understand that various aspects of this specification may be illustrated and described through several patentable categories or situations, including any new and useful processes, machines, products, or combinations of materials, or any new and useful improvements. Accordingly, all aspects of this specification can be performed entirely by hardware, entirely by software (including firmware, resident software, microcode, etc.), or a combination of hardware and software. The above hardware or software can be referred to as "data block", "module", "engine", "unit", "component", or "system". Additionally, aspects of this specification may be embodied as a computer product located in one or more computer-readable media, and the product includes computer-readable program code.

[0093] The computer storage medium may include a propagated data signal that includes computer program code, for example, on a baseband or as part of a carrier. The propagated signal may have multiple representation forms, including an electromagnetic form, an optical form, etc., or a suitable combination form. The computer storage medium can be any computer-readable

medium other than a computer-readable storage medium. The medium can be connected to an instruction execution system, apparatus, or device to implement communication, propagation, or transmission of a program for use. The program code on the computer storage medium can be propagated through any suitable media, including radio, cables, fiber optic cables, RF, or similar media, or any combination thereof.

[0094] Computer program code needed for each part of operations in this specification can be written in any one or more program languages, including object-oriented programming languages such as Java, Scala, Smalltalk, Eiffel, JADE, Emerald, C++, C#, VB.NET, and Python, conventional programmed programming languages such as C, Visual Basic, Fortran 2003, Perl, COBOL 2002, PHP, and ABAP, dynamic programming languages such as Python, Ruby, and Groovy, or other programming languages. The program code can run only on a user computer, or run on a user computer as an independent software package, or run partially on a user computer and partially on a remote computer, or run only on a remote computer or server. In the latter case, the remote computer can be connected to a user computer in any form of network, such as a local area network (LAN) or a wide area network (WAN), or connected to an external computer (for example, via the Internet), or located in a cloud computing environment, or used as a service such as software as a service (SaaS).

[0095] Furthermore, unless explicitly stated in the claims, the order of processing elements and sequences described in this specification, the use of numbers and letters, or the use of other names are not intended to limit the order of the procedures and methods of this specification. Although the above disclosure discusses through various examples what is currently considered to be a variety of useful embodiments of this specification, it is to be understood that such details are solely for that purpose, and that the appended claims are not limited to the disclosed embodiments, but, on the contrary, are intended to cover modifications and equivalent arrangements that are within the spirit and scope of the embodiments of this specification. For example, while the system components described above can be implemented by a hardware device, they can also be implemented by a software-only solution, such as installing the described system on an existing server or mobile device.

[0096] Similarly, it should be appreciated that in the above-mentioned description of embodiments of this specification, various features are sometimes incorporated into one embodiment, accompanying drawing, or description thereof for the purpose of streamlining the

descriptions disclosed in this specification and aiding in the understanding of one or more embodiments of this specification. However, this disclosure method does not mean that this specification object needs more features than the features mentioned in the claims. Rather, features of an embodiment are less than all features of a single embodiment disclosed above.

5 **[0097]** In some embodiments, numbers describing the quantities of components and attributes are used. It should be understood that such numbers used in the description of the embodiments use the modifiers "about", "approximately", or "substantially" in some examples. Unless stated otherwise, "about", "approximately", or "substantially" means that a variation of $\pm 20\%$ is allowed for the stated number. Accordingly, in some embodiments, the numerical parameters set forth in
10 the specification and claims are approximations that may vary depending upon the desired characteristics of individual embodiments. In some embodiments, numerical parameters should consider the specified significant digits and adopt the general digit retention method. Notwithstanding that the numerical fields and parameters used in some embodiments of this specification to confirm the breadth of their ranges are approximations, in specific embodiments,
15 such numerical values are set as precisely as practicable.

[0098] The entire content of each patent, patent application, patent application publication, and other materials, such as an article, a book, a specification, a publication, document, etc., cited in this specification are hereby incorporated into this specification by reference. Application history documents that are inconsistent with or conflict with the content of this specification are
20 excluded, as are documents (currently or hereafter appended to this specification) that limit the broadest scope of the claims of this specification. It should be noted that if there is any inconsistency or conflict between the descriptions, definitions, and/or terms used in the attached materials of this specification and the content of this specification, the descriptions, definitions, and/or terms used in this specification shall prevail.

25 **[0099]** At last, it should be understood that the embodiments described in this specification are merely illustrative of the principles of the embodiments of this specification. Other variants may also fall within the scope of this specification. Thus, by way of example but not of limitation, alternative configurations of the embodiments of this specification may be utilized in accordance with the teachings herein. Accordingly, the embodiments of this specification are not limited to the
30 embodiments explicitly introduced and described in this specification.

CLAIMS

What is claimed is:

1. A knowledge graph construction method, wherein the method comprises the following:

determining an underlying graph based on flow data, wherein the underlying graph comprises
5 entity nodes and an edge, and the edge comprises a flow edge that reflects data's flow relationship
between the entity nodes;

determining, based on the underlying graph, one or more higher-layer graphs having an
upper-lower hierarchical relationship, wherein the entity nodes of a lower-layer graph are
subordinate to the entity nodes of an upper-layer graph; and

10 storing the underlying graph and the higher-layer graph for query.

2. The method according to claim 1, wherein the determining, based on the underlying graph,
one or more higher-layer graphs having an upper-lower hierarchical relationship comprises the
following:

determining the entity nodes of the upper-layer graph based on labels corresponding to the
15 entity nodes of the lower-layer graph; and

determining an upper-layer flow edge based on flow endpoints of a lower-layer flow edge
and the entity nodes, of the upper-layer graph, to which the flow endpoints are subordinate,
wherein the lower-layer flow edge is the flow edge of the lower-layer graph, and the upper-layer
flow edge is the flow edge of the upper-layer graph.

20 3. The method according to claim 2, wherein the method further comprises the following:

determining an edge feature of an upper-layer flow edge based on a lower-layer flow edge
and a subordinate relationship between the lower-layer graph and the upper-layer graph, wherein
endpoints of the lower-layer flow edge comprise a first lower-layer endpoint and a second lower-
layer endpoint, endpoints of the upper-layer flow edge comprise a first upper-layer endpoint and a
25 second upper-layer endpoint, the first lower-layer endpoint is subordinate to the first upper-layer
endpoint, and the second lower-layer endpoint is subordinate to the second upper-layer endpoint.

4. The method according to claim 1, wherein the method further comprises the following:

determining, based on the lower-layer graph, a lower-layer controlled node that satisfies a
control condition;

30 determining that the entity node, of the upper-layer graph, to which the lower-layer controlled

node is subordinate is an upper-layer controlled node; and

determining a control feature of the upper-layer controlled node based on the control condition.

5 5. The method according to claim 1, wherein the flow data comprise data of funds, the entity nodes of the underlying graph correspond to sub-accounts for storing the funds, and the entity nodes of the higher-layer graph correspond to accounts or users to which the sub-accounts are subordinate.

6. A knowledge graph construction system, wherein the system comprises the following:

10 an underlying-graph determining module, configured to determine an underlying graph based on flow data, wherein the underlying graph comprises entity nodes and an edge, and the edge comprises a flow edge that reflects data's flow relationship between the entity nodes;

15 a higher-layer-graph determining module, configured to determine, based on the underlying graph, one or more higher-layer graphs having an upper-lower hierarchical relationship, wherein the entity nodes of a lower-layer graph are subordinate to the entity nodes of an upper-layer graph; and

 a graph storage module, configured to store the underlying graph and the one or more higher-layer graphs for query.

7. A data query method, wherein the method comprises the following:

20 obtaining a query request, wherein the query request comprises a query graph; and obtaining, based on the query graph and from the higher-layer graph obtained by using the method according to any one of claims 1 to 5, sub-graph data used to generate a query result.

8. A data query system, wherein the system comprises the following:

25 a query request acquisition module, configured to obtain a query request, wherein the query request comprises a query graph; and a query result acquisition module, configured to obtain, based on the query graph and from the higher-layer graph obtained by using the method according to any one of claims 1 to 5, sub-graph data used to generate a query result.

30 9. A knowledge graph construction apparatus, comprising a processor, wherein the processor is configured to perform the knowledge graph construction method according to any one of claims 1 to 5.

10. A computer-readable storage medium, wherein the storage medium stores a computer

instruction, and after a computer reads the computer instruction in the storage medium, the computer performs the knowledge graph construction method according to any one of claims 1 to 5.