

TITLE OF THE INVENTION: METHOD FOR GRAPH-BASED DIALOGUE FLOW CONFIGURATION

TECHNICAL FIELD

[0001] The present disclosure relates to graph-based dialogue flow configuration for implementing voice dialogue robots or automated voice assistants.

BACKGROUND

[0002] Artificial Intelligence (AI), with voice robots as an emerging technology in recent years, has demonstrated broad application prospects in various fields such as customer service, education, entertainment, and more. The application prospects of AI voice dialogue robot technology are vast. This technology primarily relies on preconfigured dialogue flows to achieve natural and smooth voice interactions. However, in practical applications, due to increasing complexity of businesses, the number and complexity of dialogue flows required for AI voice dialogue robots have escalated rapidly.

[0003] Currently, there are two methods for configuring dialogue process for AI voice robots: text-based dialogue process configuration and graph-based dialogue process configuration.

[0004] Text-based dialogue process configuration requires users to write extensive text scripts to describe the robot's interaction process. This not only demands high programming skills but also struggles to achieve complex voice interactions.

[0005] Graph-based dialogue process configuration technology is easier for users to operate as it allows users to configure dialogue flows for AI voice robots through visual operations such as dragging and connecting dialogue nodes in a graphical user interface. However, existing graph-based dialogue flow configuration techniques are still inadequate when dealing with the configuration of a large number of dialogue nodes. Users often find it challenging to quickly and efficiently configure high-quality dialogue flows that meet their needs from numerous and disorganized dialogue flow nodes in the configuration interface. This issue is particularly pronounced when users need to configure more complex voice interaction dialogue flows. Furthermore, current graph-based dialogue process configuration technology often lacks flexibility which is unfavorable for users to subsequently edit and optimize complex dialogue flows.

[0006] Accordingly, there is an urgent need for an improved dialogue process configuration solution for voice robots.

SUMMARY

[0007] In an aspect of the disclosure, a graph-based dialogue flow configuration method is provided and comprises:

displaying nodes in a graph-based dialogue flow editing area, wherein each node includes node attribute and node content;

applying node association rules to a target node and at least some of the displayed nodes along an upstream and/or a downstream direction of the target node to identify the target node and associated nodes as forming a flow path, wherein the displayed nodes include the target node and the associated nodes; and

determining a valid status of the flow path in respect of the target node by determining two of the associated nodes include start-type attribute and end-type attribute respectively.

[0008] In an embodiment, the step of applying the node association rules to the target node and the at least some of the displayed nodes to determine the target node and the associated nodes as forming the flow path includes:

performing a first iteration of applying the node association rules to the target node to identify a first one of the associated nodes as forming the flow path; and

performing iterations of applying the node association rules to an iteration-dependent node to progressively identify a remainder of the associated nodes as forming the flow path, wherein the iteration-dependent node for a current one of the iterations is an associated node identified in a previous one of the iterations as forming the flow path.

[0009] In an embodiment, the step of performing the first iteration of applying the node association rules to the target node to identify the first one of the associated nodes as forming the flow path includes:

determining relative position information of spatially-adjacent nodes, of the displayed nodes, with respect to the target node;

based on a proximity criteria of the relative position information and the node association rules, identifying the first one of the spatially-adjacent nodes as forming the flow path.

[0010] In an embodiment, the relative position information includes relative distances between the spatially-adjacent nodes and the target node, or coordinate information of the spatially-adjacent nodes and the target node, and wherein the proximity criteria includes closest proximity to the target node, or location within a predetermined radius from the target node.

[0011] In an embodiment, the dialogue flow editing area includes a start-type editing area and an end-type editing area,

[0012] wherein the step of determining the two of the associated nodes includes start-type node attribute and end-type node attribute respectively is based on determining the two of the associated nodes as located in the start-type editing area and the end-type editing area respectively.

[0013] In an embodiment, the method further comprises:

determining, for each node of the flow path, a count of completed node contents against a predetermined count of configurable node contents to produce a content completion status;

displaying, in the dialogue flow editing area, the content completion status of each node.

[0014] In an embodiment, the method further comprises: displaying, in the dialogue flow editing area, the valid status of the flow path.

[0015] In an embodiment, the method further comprises:

performing cross-path editing on the flow path and a valid second flow path having second associated nodes including:

in response to a copy or cut operation of a segment of the flow path, temporarily storing segment nodes of the segment, and node attributes and node contents of the segment nodes, wherein the segment nodes include end-of-segment nodes;

in response to a paste or move operation of the segment into the second flow path, determining one of the second associated nodes to be spliced with the segment nodes by:

determining second relative position information of second spatially-adjacent nodes, of the second associated nodes, with respect to the end-of-segment nodes;

based on a proximity criteria of the second relative position information, determining one of the second spatially-adjacent nodes as the one of the second associated nodes to be spliced with the segment nodes;

based on determining one of the end-of-segment nodes and the second node are compliant with the node association rules, splicing the one of the end-of-segment nodes of the first flow path to the second node of the second flow path to form a third flow path; and

displaying, in the flow editing area, at least the third flow path.

[0016] In an embodiment, the second relative position information includes relative distances between the second spatially-adjacent nodes and the end-of-segment nodes, or coordinate information of the second spatially-adjacent nodes and the end-of-segment nodes.

[0017] In an embodiment, the node association rules include node attribute association rules based on compatible node attributes and node content logic association rules based on compatible node content.

[0018] In an embodiment, the method further comprises:

based on at least one historical flow path, and node attributes and node contents of historical nodes of the at least one historical flow path, generating the node association rules.

[0019] In an aspect of the disclosure, a non-transitory computer-readable medium comprising computer-executable instructions configured to direct at least one computer processor to perform the method according to any one of the preceding embodiments is provided.

[0020] In an aspect of the disclosure, an apparatus for configuring graph-based dialogue flow is provided and comprises:

at least one memory for storing computer-executable instructions;

at least one display device; and

at least one processor communicably coupled to the at least memory and at least one display device, and configured to:

execute the computer-executable instructions to perform the method according to any one of the preceding embodiments.

BRIEF DESCRIPTION OF DRAWINGS

Embodiments of the invention will be described in detail with reference to the accompanying drawings, in which:

[0021] Figure 1 shows a dialogue flow editing area according to an embodiment;

[0022] Figure 2A shows a method for configuring dialogue flow according to an embodiment;

[0023] Figure 2B shows a method for determining a flow path according to an embodiment;

[0024] Figure 2C shows a method for determining a flow path according to an embodiment;

[0025] Figure 2D shows a method for performing editing of a flow path according to an embodiment;

[0026] Figure 3 shows a dialogue flow editing area according to an embodiment;

[0027] Figure 4 shows a dialogue flow editing area according to an embodiment; and

[0028] Figure 5 shows a dialogue flow editing area according to an embodiment.

DETAILED DESCRIPTION

[0029] In the following description, numerous specific details are set forth in order to provide a thorough understanding of various illustrative embodiments of the invention. It will be understood, however, to one skilled in the art, that embodiments of the invention may be practiced without some or all of these specific details. It is understood that the terminology used herein is for the purpose of describing particular embodiments only, and is not intended to limit the scope of the invention. In the drawings, like reference labels or numerals refer to same or similar functionalities or features throughout the several views. In the drawings, directional arrows shown between features illustrate data or information transfer therebetween in accordance with description of some embodiments but are not limited as such. In other words, data or information transfer in reverse to the directional arrows and/or not shown by directional arrow among features may be envisaged and are not shown to avoid obscuring description of the embodiments.

[0030] Embodiments described in the context of one of the apparatuses or methods are analogously valid for the other apparatuses or methods. Similarly, embodiments described in the context of an apparatus are analogously valid for a method, and vice versa.

[0031] Features that are described in the context of an embodiment or example may correspondingly be applicable to the same or similar features in the other embodiments or examples. Features that are described in the context of an embodiment or example may correspondingly be applicable to the other embodiments or examples, even if not explicitly described in these other embodiments or examples. Furthermore, additions and/or combinations and/or alternatives as described for a feature in the context of an embodiment or example may correspondingly be applicable to the same or similar feature in the other embodiments or examples.

[0032] It should be understood that the articles "a", "an" and "the" as used with regard to a feature or element include a reference to one or more of the features or elements. The term "and/or" includes any and all combinations of one or more of the associated feature or element. The terms "comprising", "including", "having", "involving" and any of their related terms, as used in description and claims, are intended to be open-ended and mean that there may be additional features or elements other than the listed ones. Identifiers such as "first", "second", "third", and so on, are used merely as labels, and are not intended to impose numerical requirements on their objects, nor construed in a manner imposing any relative position or time sequence between limitations.

[0033] The term “node” includes a reference to “script node” and may be used interchangeably. The term “determining” and its related terms may be used interchangeably with the term “identifying” and its related terms.

[0034] Figure 1 shows a non-limiting graph-based user interface for visually configuring dialogue flow. A graphical user interface may include a dialogue or script flow editing area, e.g., rectangular area, in which multiple script nodes (shown as circles) as well as their respective node attributes (shown as boxes linked to the nodes) may be displayed. A user may perform various visual operations on the nodes in the dialogue flow editing area. Non-limiting visual operations include adding new script nodes in the dialogue flow editing area, selecting/deleting existing nodes, moving/dragging existing nodes, etc. Each node includes a node attribute, including but not limited to start-type, jump-type, request-type, reinforcement-type, response-type, and end-type (termination-type), etc. Each node also includes a node content. The node attributes and node contents may be configured, e.g., edited, reset, by a user.

[0035] Figures 2A to 2D show flowcharts relating to graph-based dialogue flow configuration methods according to some embodiments. The flowcharts will be described with reference to graph-based script flow interfaces of Figures 1, 3, 4 and 5 as follows.

[0036] In the flowchart 20 of Figure 2A, block 21 includes displaying a plurality of nodes in a graph-based dialogue flow editing area. Block 21 may include displaying node attributes and/or node contents, e.g., within or outside the flow editing area, in association with the displayed nodes. The displayed nodes may form one or more node flow paths.

[0037] Block 22 of Figure 2A relates to tracing along an upstream direction and/or a downstream direction of a target node to identify a script flow path which includes at least some of the displayed nodes. In particular, block 22 includes applying node association rules to a target node and at least some of the displayed nodes along an upstream and/or a downstream direction of the target node to identify the target node and associated nodes as forming a flow path. In other words, the associated nodes which form a flow path together with the target node are selectively identified from the displayed nodes based on iterative applications of node association rules starting from the target node and progressing to other nodes along the upstream and/or downstream direction.

[0038] An example of block 22 is shown in blocks 221, and 222 of Figure 2B. Using the target node as a starting point, associated upstream nodes are progressively determined, i.e., identified, according to node association rules until there is no further associated upstream node. Each identified associated upstream node serves as the starting point for the next determination or iteration. Associated downstream nodes are also progressively determined, i.e., identified, according to node association rules until there is no further associated downstream node. Each identified associated downstream node serves as the starting point

for the next determination or iteration. In other words, block 22 may include block 221 which includes performing a first iteration of applying node association rules to the target node to identify a first one of the associated nodes. Block 22 may include block 222 which includes, using the identified first one of the associated nodes, performing a second iteration of applying the node association rules to the first one of the associated nodes to identify a second one of the associated nodes. Block 222 may further include performing a third iteration of applying node association rules to the second one of the associated nodes. This means that, after performing the first iteration in block 221, block 222 includes successively performing iterations of applying the node association rules to an iteration-dependent node to progressively identify a remainder of the associated nodes as forming the flow path. The iteration-dependent node, e.g., first one of the associated nodes, for a current one of the iterations, e.g., second iteration, is an associated node e.g., first one of the associated nodes, identified in a previous one of the iterations, e.g. first iteration. The iteration-dependent node, e.g., second one of the associated nodes, for a current one of the iterations, e.g., third iteration, is an associated node e.g., second one of the associated nodes, identified in a previous one of the iterations, e.g., second iteration. Such iterations are performed in the upstream and downstream directions of the target node until application of the node association rules does not identify new upstream or downstream associated nodes. The flow chart of Figure 2B may be performed in the upstream and/or downstream directions and may take place sequentially in any order or simultaneously.

[0039] It is to be appreciated that the target node may form part of one or multiple flow paths; block 22 may be performed for other target nodes to identify other flow paths from the displayed nodes.

[0040] Node association rules may include node attribute association rules and/or node content logic association rules. Node attribute association rules require successive or consecutive nodes to possess compatible node attributes in order for them to be allowably associated. For example, a node having node attribute "response node" is compatible with an upstream node having node attributes such as "request node" or "reinforcement node" but is not compatible with an upstream node having other node attributes. Node content logic association rules require successive or consecutive nodes to possess compatible node contents in order for them to be allowably associated. For example, a node having script content "whether to transfer to human service" is compatible with a downstream node having prescribed node contents "about to transfer you to human service" or "transferring" but is not compatible with a downstream node having other or non-prescribed node contents. Node content logic association rules may be generated based on a user's historical records of editing script flows. Accordingly, the flow chart 20 may further include the following steps (not shown): acquiring a script flow history record set, where the script flow history record set

includes at least one historical script flow path, and node attributes and node contents of historical nodes of the at least one historical flow path; generating the node association rules based on the script flow history record set. Node association rules may be predetermined and/or periodically updated.

[0041] Blocks 221 and 222 of Figure 2B may be illustrated by Figure 1 which shows a target node X (may be shown as a shaded circle) and multiple other nodes C, D, E, F, etc., (may be shown as unshaded circles) in the dialogue flow editing area. Starting from the target node X, upstream nodes A and B and downstream nodes C and D which are directly associated with the target node X are determined according to node association rules. For example, according to the node attribute association rules, node attribute "jump node" of the target node X is compatible with the node attribute "request node" of node A and, furthermore, node script content "request to transfer to human service" of target node X and node script content "about to transfer to human service" of node A have a compatible logical relationship in script content. Since node A and target node X are compliant with predetermined node association rules, node A is identified as an upstream node of target node X. Similarly, upstream node B and downstream nodes C and D which are directly associated with the target node X may be determined. Subsequently, using the identified nodes A, B, C, and D as starting points, the node association rules are repeatedly applied to determine that an upstream node A₁ is associated with node A, an upstream node B₁ is associated with node B, downstream node C₁ is associated with node C, and downstream node D₁ is associated with node D, respectively. Subsequently, using the identified nodes A₁, B₁, C₁, and D₁, identification of further subsequent upstream and/or downstream nodes are identified by applying node association rules. Each application of node association rules is iteratively performed on the node identified in the previous iteration. The upstream and/or downstream nodes which are identified as being compliant with the node association rules, together with the target node, therefore form a script flow path.

[0042] Another example of block 22 is shown in blocks 221 and 222 of Figure 2C in which block 221 includes blocks 2211 and 2212. In block 2211, relative position information of spatially-adjacent nodes may be determined. Spatially-adjacent nodes generally refer to at most some of the displayed nodes which are adjacent to the target node. In block 2212, in a first iteration, based on a proximity criteria of the relative position information and node association rules, a first or corresponding one of the spatially-adjacent nodes may be determined as a first one of the associated nodes being directly associated with the target node and forming a flow path with the target node. Block 2212 then proceeds to block 222, which is similar to block 222 of Figure 2B, where second and possibly subsequent iterations are performed based on node association rules and without considering the relative position information to identify a second and possibly subsequent ones of the associated nodes which

form the flow path. The flow chart of Figure 2C may be performed in the upstream and/or downstream directions and may take place sequentially in any order or simultaneously.

[0043] The relative position information may be relative distances between the spatially-adjacent nodes and the target node, or coordinate information of the spatially-adjacent nodes and the target node, or other suitable parameters. The proximity criteria may be based on closest proximity to the target node, e.g., shortest relative distance, or location within a predetermined radius from the target node, e.g., within a threshold distance of the target node, or other suitable criteria.

[0044] Blocks 2211 and 2212 may be illustrated by Figure 3 which shows a similar arrangement of displayed nodes as Figure 1. Relative position information of nodes near the target node X, including the position coordinates of nearby nodes A, B, C, D, E, and F, may be obtained. Distances d_1 , d_2 , d_3 , d_4 , d_5 , d_6 from nodes A, B, C, D, E, and F, respectively, to the target node X are computed. The relationships of the distances are derived as $d_6 > d_4 > d_5 > d_2 > d_3 > d_1$, where d_6 is the longest relative distance and d_1 is the shortest relative distance. Subsequently, based on node association rules between the target node X and nodes A, B, C, D, E, and F, as well as a proximity criteria of relative distances, a first upstream associated node is determined from nodes A, B, C, D, E, and F. For the upstream nodes A, B, and E of the target node X, node attribute "request node" of node A and node attribute "reinforcement node" of node B are compatible with node attribute "jump node" of the target node X, while node attribute of node E and the target node X are not compatible, hence node E is excluded from any flow path involving target node X. As distances between nodes A and B and the target node X are d_1 and d_2 , respectively, and d_2 is greater than d_1 ($d_2 > d_1$), node A is determined as the first upstream node being directly associated with the target node. Similarly, node C can be determined as the first downstream node directly associated with the target node X. Subsequently, without considering the relative distances between the target node X and nodes A, B, C, and D, a second upstream node A_1 is determined as directly associated with the first upstream node A and the second downstream node C_1 is determined as directly associated with the first downstream node C based on node association rules. This application of node associate rules is repeated to determine a second upstream node A_2 as directly associated with node A_1 and a second downstream node C_2 is determined as directly associated with node C_1 , respectively. This application of node association rules is repeated on successive nodes until no new upstream or downstream associated node can be further found.

[0045] In an alternative example, the proximity criteria of relative distances may be based on a threshold distance R between nodes (see the dashed-line circle with a radius of R in Figure 3). In particular, based on the distances d_1 , d_2 , d_3 , d_4 , d_5 , d_6 from nearby nodes A, B, C, D, E, and F, respectively, to the target node X, the relationships of the distances are derived as d_4

$> R$ and $d_6 > R$, while $d_1 < R$, $d_2 < R$, $d_3 < R$, and $d_5 < R$. Thus, based on node association rules and the relative position information between the target node X and nodes A, B, C, D, E, and F, a first upstream node is determined from nodes A, B, C, D, E, and F. For the upstream node E, since its distance d_6 to the target node X is greater than the threshold R, node E is excluded from any flow path involving target node X. Distances between nodes A and B and the target node X are d_1 and d_2 , respectively, and both $d_1 < R$ and $d_2 < R$. At the same time, node attribute "request node" of node A and the node attribute "reinforcement node" of node B are compatible with the node attribute "jump node" of the target node X; node content "about to transfer to an agent" of node A, node content "confirm transfer" of node B, and node content "request to transfer to an agent" of the target node X are also compatible. Therefore, nodes A and B are determined as first upstream nodes directly associated with the target node X. Similarly, nodes C and D can be determined as first downstream nodes directly associated with target node X. Subsequently, without considering the relative distances between the target node X and nodes A, B, C, and D, second upstream/downstream nodes associated with nodes A, B, C, and D are determined based only on node association rules, until no new upstream or downstream associated node can be found.

[0046] Block 23 of Figure 2A includes determining a valid status of the identified flow path in respect of the target node by determining a presence of start-type attribute and end-type attribute in two of the associated nodes respectively. On the other hand, if an absence of start-type attribute and/or end-type attribute is determined in the associated nodes of the flow path, an invalid status is determined for the flow path.

[0047] In an embodiment, block 23 may include determining whether a node attribute of any one of the nodes of the flow path is start-type and whether a node attribute of another one of the nodes of the flow path is end-type. If two nodes of the flow path are determined as start-type and end-type respectively, the flow path is determined as having a valid status, i.e., the flow path is valid, in respect of the target node. The two nodes may be final nodes at opposing endpoints of the flow path, i.e., end-of-path nodes, or non-final nodes of the flow path, i.e., intermediate nodes, or a combination of a final node and a non-final node.

[0048] Alternatively, block 23 may include determining whether a node attribute of any one of the nodes of the flow path is start-type and whether a node attribute of another one of the nodes of the flow path is end-type by determining whether any two nodes of the flow path are respectively located in a start-type editing area and an end-type editing area of the flow editing area. This alternative is illustrated in Figure 4 in which the dialogue flow editing area includes a start-type editing area and an end-type editing area, where nodes displayed therein have start-type attributes and end-type attributes respectively. When a node is positioned in the start-type editing area, its node attribute may be defined as or modified to start-type by default. When a node is positioned in the end-type editing area, its node attribute may be defined as

or modified to end-type by default. Referring to a flow path $B_2 \rightarrow B_1 \rightarrow B \rightarrow X \rightarrow D \rightarrow D_1 \rightarrow D_2$ in Figure 4, endpoint nodes of the flow path are nodes B_2 and D_2 . Node B_2 of Figure 4 is located in the start-type editing area, hence its node attribute is modified to a start-type; node D_2 of Figure 4 is located in the end-type editing area, hence its node attribute is modified to end-type. Therefore, the script flow path $B_2 \rightarrow B_1 \rightarrow B \rightarrow X \rightarrow D \rightarrow D_1 \rightarrow D_2$ of Figure 4 is determined as a valid flow path in respect of the target node X .

[0049] Block 24 of Figure 2A, which may be optional, includes determining content completion status of each node of a flow path. In particular, completion status of script contents of a node may be determined by determining a count of completed node contents against a predetermined count of configurable node contents to produce a content completion status. This determination may include comparing both counts and deriving the completion status based on the compared counts. Non-limiting examples of content completion status include a content completion percentage figure (see Figure 4), a binary completion status (not shown) such as “completed” or “pending”. Non-limiting examples of the display of content completion status include displaying the completion status within or outside the flow editing area, separate from or integral with node representation. In Figure 4, for simplification purpose, content completion status is displayed for some of the nodes, i.e., A, B, C, D, of a valid flow path using marking boxes which are located within the dialogue flow editing area and connected to the nodes.

[0050] Block 25 of Figure 2A includes displaying, in the flow editing area, a valid or an invalid status of a flow path according to the outcome of block 23. Block 25 may include displaying content completion status of each node of a flow path.

[0051] Block 25 may be illustrated by Figure 1 which shows two script flow paths which include target node X . A first flow path defined by a node sequence of $A_2 \rightarrow A_1 \rightarrow A \rightarrow X \rightarrow C \rightarrow C_1 \rightarrow C_2$ includes nodes A_2 and C_2 at its endpoints. Upon verification, node A_2 is a start node, i.e., having start-type node attribute, and node C_2 is an end node, i.e., having end-type node attribute. Therefore, the first flow path having the node sequence of $A_2 \rightarrow A_1 \rightarrow A \rightarrow X \rightarrow C \rightarrow C_1 \rightarrow C_2$ is identified as a valid flow path for target node X . This valid status may be represented in the flow editing area as solid line connections among the nodes of the first flow path. On the other hand, a second flow path defined by a node sequence of $B_2 \rightarrow B_1 \rightarrow B \rightarrow X \rightarrow D \rightarrow D_1 \rightarrow D_2$ includes nodes B_2 and D_2 at its endpoints. Upon verification, node B_2 , having a node attribute of a jump node, is identified as not having start-type attribute, while node D_2 , having node attribute of a reply node, is identified as not having end-type attribute. Therefore, the second flow path having the node sequence $B_2 \rightarrow B_1 \rightarrow B \rightarrow X \rightarrow D \rightarrow D_1 \rightarrow D_2$ in this order, is not a valid flow path for target node X . This invalid status may be represented in the flow editing area as dashed line connections among nodes of the second flow path.

[0052] Block 26 of Figure 2A relates to performing cross-path editing operations on multiple valid flow paths. This cross-path editing procedure is described for two valid flow paths as follows. The two valid flow paths may be referred to as a first flow path having first associated nodes, and a second flow path having second associated nodes. Block 26 may include blocks 261 and 262 of Figure 2D.

[0053] In block 261, a segment of the first flow path is determined for a copy or cut operation. The segment includes segment nodes which include end-of-segment nodes located at opposed endpoints of the segment. In response to the copy or cut operation, the segment to be spliced is temporarily stored. In particular, the segment nodes as well as their node attributes and node contents are temporarily stored, e.g., copied or cut together.

[0054] In block 262, the copied or cut segment of the first flow path is determined for a paste or move operation into the second flow path. In response to the paste or move operation, one of the second associated nodes of the second flow path is determined to be spliced with the copied or cut segment nodes. This determination may include block 2621 of Figure 2D which includes determining second relative position information of second spatially-adjacent nodes of the second flow path with respect to the copied or cut end-of-segment nodes of the first flow path. Second spatially-adjacent nodes of the second flow path refer to some of the second associated nodes of the second flow path that are near the end-of-segment nodes. This determination may take place when the copied or cut segment is positioned near to the second flow path. Block 2621 proceeds to block 2622 of Figure 2D in which, based on a proximity criteria of the second relative position information, one of the second spatially-adjacent nodes may be determined as the one of the second associated nodes to be spliced with the copied or cut segment nodes.

[0055] Block 2623 of Figure 2D determines whether one of the copied or cut end-of-segment nodes and the one of the second associated nodes as identified in block 2622 comply with node association rules. If it is determined that they are compliant with node association rules, the segment, i.e., the one of the copied or cut end-of-segment nodes, is spliced to the second flow path, i.e., the one of the second associated nodes identified in block 2622, to produce a third flow path. After splicing, the third flow path comprises a spliced combination of the second flow path and the copied or cut segment of the first flow path. If it is determined that they are not compliant, the paste or move operation is rejected, i.e., the copied or cut segment is not spliced to the second flow path.

[0056] The second relative position information may be second relative distances between the second spatially-adjacent nodes and the end-of-segment nodes, or coordinate information of the second spatially-adjacent nodes and the end-of-segment nodes.

[0057] Block 27 of Figure 2A includes displaying, in the flow editing area, the edited flow paths according to the outcome of block 2623.

[0058] Block 26 of Figure 2A, blocks 261, 262, 2621, 2622, 2623 of Figure 2D, and block 27 of Figure 2A may be illustrated by Figure 5 which displays two valid flow paths in the dialogue flow editing area. The two flow paths include a first flow path formed by nodes $B_2 \rightarrow B_1 \rightarrow B \rightarrow X \rightarrow C \rightarrow C_1 \rightarrow C_2$, and a second flow path formed by nodes $A_2 \rightarrow A_1 \rightarrow A \rightarrow Y \rightarrow D \rightarrow D_1 \rightarrow D_2$. At this time, a segment comprising nodes $C \rightarrow C_1 \rightarrow C_2$ of the first flow path is selected for a copy operation. The segment may be moved to a lower right corner of the dialogue flow editing area. Based on a relative position information which is determined for nodes of the second flow path that are spatially- adjacent to end-of-segment nodes of the segment, e.g., node C of the segment, node D is identified as an associated node of the second flow path which is to be spliced with the segment. Subsequently, it is determined that the node attribute "request node" of node D is compatible with the node attribute "jump node" of endpoint node C, which complies with node attribute association rules. Moreover, it is determined the script content "request to transfer to an agent" of node D is logically compatible with node attribute "about to transfer to an agent" of endpoint node C, which complies with node content logic association rules. Therefore, based on this determination that the node association rules are complied with, node D is spliced with endpoint node C to produce a new path having valid status, e.g., a third flow path which is defined by associated nodes $A_2 \rightarrow A_1 \rightarrow A \rightarrow Y \rightarrow D \rightarrow C \rightarrow C_1 \rightarrow C_2$. The splice may be indicated by a double line connection (as shown in Figure 5).

[0059] It is to be appreciated that the flow charts showing logic flows are representative of exemplary methodologies for performing novel aspects of the invention. While, for purposes of simplicity of explanation, the one or more methodologies shown herein are shown and described as a series of acts, those skilled in the art will understand and appreciate that the methodologies are not limited by the order of acts. Some acts may, in accordance therewith, occur in a different order and/or concurrently with other acts from that shown and described herein. For example, those skilled in the art will understand and appreciate that a methodology could alternatively be represented as a series of interrelated states or events, such as in a state diagram. Moreover, not all acts illustrated in a methodology may be required for a novel implementation.

[0060] The flow charts showing logic flow may be implemented in software, firmware, hardware, or any combination thereof. In software and firmware embodiments, the logic flow may be implemented by computer executable instructions or code stored on a non-transitory computer-readable medium or machine readable medium, such as an optical, magnetic or semiconductor storage. The computer-executable instructions are configured to direct at least one computer processor to perform the logic flow. The embodiments are not limited in this context.

[0061] It is to be appreciated that an apparatus for configuring graph-based dialogue flow may comprise at least one memory, at least one display device, and at least one processor communicably coupled to the memory and the display device. The processor may be any type of computer processor, such as a Central Processing Unit (CPU), a Graphics Processing Unit (GPU), a quantum processor, a machine learning processor, an artificial intelligence processor, a neural network processor, an artificial intelligence accelerator, an application specific integrated circuit (ASIC), an analog signal processor, an analog computer, a microprocessor, a digital signal processor, a field programmable gate array (FPGA), a tensor processing unit (TPU), a data processing unit (DPU), etc., or other device configured to execute code or computer-executable instructions to perform or implement the flowcharts, algorithms, processes, or operations detailed herein. The memory may be a non-transitory computer readable medium or machine readable medium for storing or comprising the code or computer-executable instructions. Examples include, but are not limited to, volatile memory, e.g., DRAM), nonvolatile memory, e.g., read-only memory (ROM)), high bandwidth memory (HBM), flash memory, solid state memory, and/or a hard drive). The display device may provide a graphical user interface for implementing visual operations in the dialogue flow editing area, including various displaying steps describing herein.

[0062] It is to be understood that the embodiments and features described above should be considered exemplary and not restrictive. Many other embodiments will be apparent to those skilled in the art from consideration of the specification and practice of the invention. Furthermore, certain terminology has been used for the purposes of descriptive clarity, and not to limit the disclosed embodiments of the invention.

CLAIMS

[Claim 1] A graph-based dialogue flow configuration method, comprising:

displaying nodes in a graph-based dialogue flow editing area, wherein each node includes node attribute and node content;

applying node association rules to a target node and at least some of the displayed nodes along an upstream and/or a downstream direction of the target node to identify the target node and associated nodes as forming a flow path, wherein the displayed nodes include the target node and the associated nodes; and

determining a valid status of the flow path in respect of the target node by determining two of the associated nodes include start-type attribute and end-type attribute respectively.

[Claim 2] The method of claim 1, wherein applying the node association rules to the target node and the at least some of the displayed nodes to determine the target node and the associated nodes as forming the flow path includes:

performing a first iteration of applying the node association rules to the target node to identify a first one of the associated nodes as forming the flow path; and

performing iterations of applying the node association rules to an iteration-dependent node to progressively identify a remainder of the associated nodes as forming the flow path, wherein the iteration-dependent node for a current one of the iterations is an associated node identified in a previous one of the iterations as forming the flow path.

[Claim 3] The method of claim 2, wherein performing the first iteration of applying the node association rules to the target node to identify the first one of the associated nodes as forming the flow path includes:

determining relative position information of spatially-adjacent nodes, of the displayed nodes, with respect to the target node;

based on a proximity criteria of the relative position information and the node association rules, identifying the first one of the spatially-adjacent nodes as forming the flow path.

[Claim 4] The method of any one of claim 1 to claim 3, wherein the relative position information includes relative distances between the spatially-adjacent nodes and the target node, or coordinate information of the spatially-adjacent nodes and the target node, and wherein the proximity criteria includes closest proximity to the target node, or location within a predetermined radius from the target node.

[Claim 5] The method of any one of claim 1 to claim 4, wherein the dialogue flow editing area includes a start-type editing area and an end-type editing area,

wherein determining the two of the associated nodes include start-type node attribute and end-type node attribute respectively is based on determining the two of the associated nodes as located in the start-type editing area and the end-type editing area respectively.

[Claim 6] The method of any one of claim 1 to claim 5, further comprising:

determining, for each node of the flow path, a count of completed node contents against a predetermined count of configurable node contents to produce a content completion status;

displaying, in the dialogue flow editing area, the content completion status of each node.

[Claim 7] The method of any one of claim 1 to claim 6, further comprising: displaying, in the dialogue flow editing area, the valid status of the flow path.

[Claim 8] The method of any one of claim 1 to claim 7, further comprising:

performing cross-path editing on the flow path and a valid second flow path having second associated nodes including:

in response to a copy or cut operation of a segment of the flow path, temporarily storing segment nodes of the segment, and node attributes and node contents of the segment nodes, wherein the segment nodes include end-of-segment nodes;

in response to a paste or move operation of the segment into the second flow path, determining one of the second associated nodes to be spliced with the segment nodes by:

determining second relative position information of second spatially-adjacent nodes, of the second associated nodes, with respect to the end-of-segment nodes;

based on a proximity criteria of the second relative position information, determining one of the second spatially-adjacent nodes as the one of the second associated nodes to be spliced with the segment nodes;

based on determining one of the end-of-segment nodes and the second node are compliant with the node association rules, splicing the one of the end-of-segment nodes of the first flow path to the second node of the second flow path to form a third flow path; and

displaying, in the flow editing area, at least the third flow path.

[Claim 9] The method of claim 8, wherein the second relative position information includes relative distances between the second spatially-adjacent nodes and the end-of-segment nodes, or coordinate information of the second spatially-adjacent nodes and the end-of-segment nodes.

[Claim 10] The method of any one of claim 1 to claim 9, wherein the node association rules include node attribute association rules based on compatible node attributes and node content logic association rules based on compatible node content.

[Claim 11] The method of any one of claim 1 to claim 10, further comprising:

based on at least one historical flow path, and node attributes and node contents of historical nodes of the at least one historical flow path, generating the node association rules.

[Claim 12] A non-transitory computer-readable medium comprising computer-executable instructions configured to direct at least one computer processor to perform the method according to any one of claim 1 to claim 11.

[Claim 13] An apparatus for configuring graph-based dialogue flow, the apparatus comprising:

at least one memory for storing computer-executable instructions;

at least one display device; and

at least one processor communicably coupled to the at least memory and at least one display device, and configured to:

execute the computer-executable instructions to perform the method according to any one of claim 1 to claim 11.