SYSTEM FOR MANAGING INFORMATION IN A MESH NETWORK

FIELD OF THE INVENTION

The present invention relates to a system for managing information in a mesh network.

BACKGROUND OF THE INVENTION

Identifiers have been used in many applications of Distributed Control Systems. In applications that connect a variety of data sources such as database, Field Control System (FCS), etc, the applications provide the information of the devices, location of the data sources as well as the relationship between the devices. The relationship between the devices can be very complex and usually they form a mesh network.

Existing systems use trees instead of mesh network to present the relationships between the devices. Tree structure can be easily managed (add, delete, search, etc.) and implemented in Relational Database Management System (RDBMS). However, vast amount of information is lost since tree hierarchy is not capable of storing all the relationships between devices. Most existing systems only store the location of the tags and some common information and uses another data structure to store the relationship (parent and child) between the tags.

Figure 1 shows the general data model of the prior art. As shown, it uses a table to store the device information including the ID, Tag name, Data, DEV ID (Device ID), etc. There is another table storing the relationship between the devices. The figure shows device with tag name TAG001 is the parent of device TAG002 and is the child of the device TAG003.

An example of the existing system is disclosed in US Patent No. 7089530. In particular, it discloses that connection objects or other such data structures facilitate establishing and configuring connections between objects that model components in a process control system. A first set of data structures (e.g. the object connection type structures) identify valid types for component-to-component pairings and the respective roles of each component in the pairing

(e.g., parent or child, source or sink). A second set of data structures (e.g., the parameter connection type structures) supply similar information for parameter-to-parameter connections. Together, these data structures can be used, for example, to validate component-to-component connections suggested by the user and to automatically configure parameter-to-parameter connections. Actual connections, both at the component or parameter level, are reflected using parameter overrides within the parameterized object model with which the connection objects are constructed.

There are some common disadvantages of such prior art systems. First of all, the cost of the search is high. Secondly, in case one device is deleted, the system needs to go the entire relationship list to find its parents and delete its reference. Thirdly, this data model is only convenient for sequential traversal, and it is difficult to perform other types of traversal. Lastly, if it allows the user to insert/delete a device, the ID will be adjusted accordingly. As a result, it needs to update all the references of the device whenever their ID has been changed.

Also, there is a need to have data structure that enables fast search speed, ease of traverse, and ease of performing common operations such as adding, deleting and updating.

It is an object of the present invention to overcome, or substantially ameliorate, the disadvantages and shortcomings of the prior art.

Other objects and advantages of the present invention will become apparent from the following description, taken in connection with the accompanying drawings, which illustrate one preferred embodiment of the invention.

SUMMARY OF THE INVENTION

In accordance with a first aspect of the invention, there is provided a system for managing information in a mesh network, the system comprising a first table for storing a plurality of device identifiers, the device identifier being mapped into an index, a second table for storing information of the device identifier, a third table for storing the relationship between the device

identifiers, the first table is being linked to the second table and the second table is being linked to the third table.

Preferably, the index is mapped into a second table by a hash function.

Preferably, the relationship between the device identifiers comprise of parent-child relationship.

In accordance with a second aspect of the invention, there is provided a method for managing information in a mesh network, the method comprising providing a system comprising a first table for storing a plurality of device identifiers, the device identifier being mapped into an index, a second table for storing information of the device identifier, a third table for storing the relationship between the device identifiers, the first table is being linked to the second table and the second table is being linked to the third table, searching the device identifier and the index of the device identifier in the first table, searching for information of the device identifier, searching for relationship of the device identifier with other device identifier.

Preferably, the index is mapped into a second table by a hash function.

Preferably, the relationship between the device identifier comprises of parent-child relationship.

Preferably, method for managing information in a mesh network further comprising of at least a searching, traverse, deletion and addition operations.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows the general data model of the prior art.

Figure 2 shows a preferred embodiment of the present invention.

Figure 3 shows a flow chart of a method of practising the invention.

DETAILED DESCRIPTION OF THE INVENTION

Figure 2 illustrates the preferred embodiment of the present invention. As shown, the system is divided into three main components. The first component, also known as a Hash Table 202, is a first table for storing a plurality of device identifiers. Each device identifier, Device ID, is mapped to a particular Index. For example, device identifier DEV001 is mapped into Index 2. The second component, also known as a Device Information Table 203, is a second table for storing information and data of the device identifier. The Hash Table 202 maps the Index into the Device Information Table by using a Hash function. As illustrated, Index 2 in Hash Table 202 is mapped to DEV001 in Device Information Table 203. The third component, also known as a Reference Table 204, is a third table for storing the relationship between the device identifiers. This table stores other relationships, such as parent and child relationship. If there is a directional relationship from node A to node B, node A is known as node B's parent and node B is known as node A's child. For example, referring to Figure 1, for DEV001, DEV002 is its child and DEV003 is its parent. In Figure 2, DEV001 is stored in the Device Information Table 203 and having a child relationship to DEV002 and having a parent relationship to DEV003.

When in use, a user first enters a query in the operating section 201. In particularly, the user could search, traverse, delete or add the query. The processor 205 would then process the search query. Thereafter, the Hash Table 202 identifies the device ID based on the search query. Each device ID is mapped to a particular index. The Hash Table 202 would map the index into the Device Information Table 203 by using an algorithm known as a Hash function. The device information table 203, is a second table for storing information and data of the device identifier. The processor 205 would then process the device information in the device information table 203. The relationship between the device identifiers which is stored in the reference table 204 is finally displayed in the Data Retrieving Section 206.

Figure 3 shows the flow chart of the system. As shown, step 301 comprises of any one of at least 4 operations, namely searching, traverse, deletion and addition.

In an example embodiment of the present invention, with reference to a device DEV001, in the first operation, "Search", the system first finds the index to the Device Information storage by using the Hash Table which is 2. The system locates the record and retrieves the information of

the device. From the reference, system knows that the device with Device ID DEV002 is its child and DEV003 is its parent. If it is necessary, system can also obtain the information of its parents and children by following the above operations again.

The second operation, “Traverse”, in step 301 shows the system may repeat the same steps and the system can traverse the whole mesh by using certain traverse algorithm of mesh.

The third operation, “Delete”, in step 301 shows that if system wants to delete a device, e.g. DEV002, system first finds the index to the Device Information storage by using the Hash table which is 1 in this example. System gets the Device ID of its children and parents. Here DEV001 is the only parent of DEV002. Then system deletes the record of DEV002 from the list. Last it locates its parent DEV001 by using the Hash Table again and removes DEV002 from its children references.

The fourth operation, “Add”, in step 301 shows that if system wants to add a new device with Device ID DEV010. It first uses the Device ID DEV010 to create a new mapping in the Hash Table and get the index to the Device Information which is 3. It locates the space according to the index, and fills up the field with the proper information of the device. If it has relationships with other devices, system adds the Device IDs of these devices into its references and specifies whether they are its parents or children. System finds its parents and children by the steps describing in the searching and add the DEV010 to their references accordingly

After the four operations in step 301, the index may be mapped into the device information table in step 302. In step 303, the information of the device is processed in the device information table before the relationship of the device identifier with other device identifier is being processed in step 304 in the reference table.

It is to be understood that the invention is not intended to be limited to the illustrated embodiment and the numerous alternatives and modifications will be apparent to those skilled in the art. Therefore, it is to be understood that the invention is to be defined only by the following claims.

Claims:

1. A system for managing information in a mesh network, the system comprising:

a first table for storing a plurality of device identifiers, the device identifier being mapped into an index,

a second table for storing information of the device identifier,

a third table for storing the relationship between the device identifiers,

the first table is being linked to the second table and the second table is being linked to the third table.

2. The system according to claim 1, wherein the index is mapped into a second table by a hash function.

3. The system according to any one of the preceding claims, wherein the relationship between the device identifiers comprise of parent-child relationship.

4. A method for managing information in a mesh network, the method comprising:

providing a system comprising a first table for storing a plurality of device identifiers, the device identifier being mapped into an index, a second table for storing information of the device identifier, a third table for storing the relationship between the device identifiers, the first table is being linked to the second table and the second table is being linked to the third table,

searching the device identifier and the index of the device identifier in the first table,

searching for information of the device identifier,

searching for relationship of the device identifier with other device identifier.

5. The method according to claim 4, wherein the index is mapped into a second table by a hash function.

6. The method according to claims 4 or 5, wherein the relationship between the device identifier comprises of parent-child relationship.

7. The method according to any one of claims 4 to 6, further comprising of at least a searching, traverse, deletion and addition operations.