A METHOD AND SYSTEM FOR VERIFYING A RULE SET FOR AN INFORMATION MODEL

FIELD OF INVENTION

This invention relates to a method and system of verifying a rule set for an information model.

BACKGROUND OF THE INVENTION

In a complex information model, for example, an OPC-Unified Architecture (OPC-UA) specifications, it is difficult to verify if the resulting information model is constructed in full compliance with the specifications. These difficulties arise from the understanding of the specifications and conveying the message from the designer to the implementer. Usually the specifications are written in English.

In addition, as a result of globalisation, a software product can be developed by parts in several countries to improve the turnaround time to develop the product. The developers or designers in different countries developing the different parts of the product may not be proficient in English language or the any other language which the specifications have been documented in. As a result of the language barrier, the understanding and interpretation of the specifications may be varied, affecting the quality of the developed product.

There is a need to provide a solution to ensure that the product is developed according to the specifications.

Further, when developing a software product, for each change to the product, the software will have to be compiled before the verification can be performed. Software developers prefer to incorporate as many changes as possible prior to running each compilation to reduce the time taken for the compilation. However, this causes much inconvenience when troubleshooting which part of the software is failing.

Therefore, there is another need to verify the software product without the need for the step of compilation.

In the prior art, US patent number 7,080,142 discloses a system and method of using rules to manage computer resources. The system and method use an event provider to indicate an event relating to a computer resource usage; an action provider to cause an action that impacts a computer resource usage; a rules database maintaining an extensible and dynamic rule set; and a rules engine coupled with the rules database to communicate with the event and action providers.

In another prior art, US patent number 7,051,243 discloses a system and method for identifying problems with a system configuration information system. The system and method use a rules engine configured to access the configuration and rules data for identifying configuration issues.

SUMMARY OF THE INVENTION

Briefly, the invention comprises a method for verifying at least one rule set for an information model, having interrelated nodes with specific relationships, including the steps of providing at least one index table of the nodes and the relationships with other nodes, providing at least one rule set, providing a rule engine, activating the rule engine to access the index table and the rule set, and thereby verifying the index table.

The method further includes the step of identifying a list of rule violations.

In addition, the invention comprises a system for verifying at least one rule set for an information model, having interrelated nodes with specific relationships, including at least one index table of the nodes and the relationships with other nodes, a database of a plurality of rule sets, a rule engine, and a reporting module to verify the index table, wherein the rule engine is activated to access the index table and the plurality of rule sets, thereby verifying the index table.

BRIEF DESCRIPTION OF THE DRAWING

Figure 1 illustrates an information model having interrelated nodes with specific relationships.

Figure 2 is a flowchart of the invention.

Figure 3 is an index table.

Figure 4 illustrates an example system of the invention.

Figure 5 illustrates a preferred system of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Figure 1 illustrates a simplified example of an information model 100 which is a mesh-network of interrelated nodes with specific relationships. In the mesh network, every node is related to at least one other node. Each node 120 in the model is unique and contains specific information. Further, each node 120 has a specific type of relationship 140 with at least one other node 120 in the model.

A first embodiment of the invention relates to a method for verifying at least one rule set for an information model 100 having interrelated nodes with specific relationships. A first embodiment of the method is illustrated in Figure 2. Before applying or using the invention, the specification for the information model design is finalised in Step 402. Based on the information model specification, the method in the invention includes the step 406 of providing at least one index table, an example as shown in Figure 3. The index table is the result of the application created by implementers of the information model 100. The application is created to capture all the semantic information of the model 100. The first column NodeID 210 lists all the nodes existing in the model. The NodeID 210 is a structured element including a unique identifier, node type, location, size, start position and any other necessary information. The second column Relationships 220 lists the type of relationship between the node list in the NodeID 210 and the other related nodes. The relationship can be forward or reverse. The next step is providing a rule set 404. Steps 404 and 406 can be performed concurrently. The rule set is created by the rule designer. Further, the steps include providing a rule engine, activating the rule engine and verifying the rule set. In the step 408 of activating the rule engine, the rule engine accesses the rule set and the index table.

In the first embodiment, the step 410 of generating a report includes the step of identifying a list of rule violations.

When a list of rule violations is identified in step 410, the list of rule violations will be generated for the user to review. The method further includes the steps of fixing the list of rule violations, and choosing an option to verify the rule.

The options elected to fix the list of rule violations will depend on the cause of the violation. If at least one rule violation is due to a mistake in the rule set created, the implementer will feedback to the rule designer to modify the rule set in step 412.

If at least one rule violation is due to the application to capture all the semantic information of the model being in error and causing error in the index table, the implementer will need to update the index table by correcting the application in step 414.

If at least one of the rule violations are not due to any of the above errors, the implementer can add one or more subsequent rule sets of their own in step 416.

After the step of fixing the rule violation, there is provided a step 418 for choosing an option whether to re-verify the rule set or sets again. When the option to re-verify is chosen, the rule engine is activated again. Otherwise, the verification process is completed, illustrated as numeral 420.

The invention allows the verification of a complex information model to be done simply and conveniently. It allows the implementers to plug and use any rule set easily without the need for compilation. Further, any errors can be identified and resolved easily since the rule sets can be implemented one at a time or separately. In addition, the rule designers can modify the rules whenever necessary. The modified rules can be easily used by the implementers. The misinterpretation of the specifications of the information model is reduced which in turn improves the accuracy of the product.

A system for operating the invention is shown in Figure 4. The system includes at least one index table 330 of the nodes and the relationships with other nodes, at least one rule set 320, a rule engine 312, and a reporting module 340 to report verification results, wherein the verification results are generated when the rule engine is activated to access the index table and the rule set.

Preferably, there are the rule sets are stored in a database 320 for ease of management. Further, the rule sets are stored in a master station. This advantageously allows other users

to access and apply the rule sets which have been previously developed for other applications.

An example of the system 300 is shown in Figure 5. In the example, an Information Model Builder application 310 is used. The application 310 can be either a real-time verifier service running in the background or an on-demand application. The application 310 has a rule engine 312. In the system, there are multiple rule sets 322, 324, 326 for grouping the rule sets by the functions. For example, a Base Rule Set 322 has rules which perform the basic functions of the application, Rule Sets 1 and 2 of reference numerals 324 and 326 respectively have been added for verification. The rule sets can be stored in a database 320.

The rule engine is activated to access to the rule sets in the database 320 and an index table 330 and begins the verification process. The rule engine can be activated manually or automatically for example at predetermined time intervals, when the index table is updated and when a rule is added to the database. When the verification process is completed, a list of rule violations will be generated for review in a reporting module 340.

Preferably, the rule set in the above discussion is encoded using eXtensible Markup Language.

We claim

1. A method for verifying a rule set for an information model, having interrelated nodes with specific relationships, including the steps of

(a) providing at least one index table of said nodes and the relationships with other nodes;

(b) providing at least one rule set;

(c) providing a rule engine;

(d) activating said rule engine to access said index table and said rule; and

(e) thereby verifying the index table.

2. A method as claimed in claim 1 wherein the step of verifying the index table includes the step of

identifying a list of rule violations.

3. A method as claimed in claim 2 wherein said rule set is encoded using eXtensible Markup Language.

4. A method as claimed in claim 2 wherein a list of rule violations is identified, further includes the steps of

fixing said list of rule violations; and

choosing an option to verify said rule,

when the option to verify said rule is chosen, the method further includes the steps of

activating said rule engine, and

thereby verifying the rule set.

5. A method as claimed in claim 4 wherein fixing said list of rule violations is performed by modifying said rule set.

6. A method as claimed in claim 4 wherein fixing said list of rule violations is performed by updating said index table.

7. A method as claimed in claim 4 wherein fixing said list of rule violations is performed by adding another rule set.

8. A method as claimed in claim 1 wherein activating said rule engine is performed manually.

9. A method as claimed in claim 1 wherein activating said rule engine is performed automatically at a predetermined time interval.

10. A method as claimed in claim 1 wherein activating said rule engine is performed automatically when said index table is updated.

11. A system for verifying one or more rule sets for an information model, having interrelated nodes with specific relationships, including

(a) at least one index table of said nodes and the relationships with other nodes;

(b) at least one rule set;

(c) a rule engine to verify the rule set; and

(d) a reporting module to report results,

wherein the index table is verified when said rule engine is activated to access said index table and said rule set.

12. A system as claimed in claim 11 wherein the rule set is encoded using eXtensible Markup Language.

13. A system as claimed in claim 11 further includes

a database for storing wherein said rule set.

14. A system as claimed in claim 11 further includes

a master station for storing said rule set for use in other information model or applications.

15. A system as claimed in claim 11 wherein said reporting module generates a list of rule violations.

16. A system as claimed in claim 11 wherein said rule engine is activated manually.

17. A system as claimed in claim 11 wherein said rule engine is activated automatically at a predetermined time interval.

18. A system as claimed in claim 11 wherein said rule engine is activated automatically when said index table is updated.

19. A system as claimed in claim 12 wherein said rule engine is activated automatically when a new rule is added to said database.