Abstract—The paper further studies on E-LOTOS. In the paper, we design various graphics styles of GE-LOTOS, which makes to express the E-LOTOS text meaning accurately. And the paper realizes conversion software from the textual E-LOTOS to GE-LOTOS, and realizes to generate test cases using GE-LOTOS. The paper gives the GE-LOTOS presentation and the TTCN-3 abstract test suite of the BGP-4's finite state machine.

Keywords—Protocol Conformance Test; GE-LOTOS; TTCN-3; Abstract Test Suite; BGP-4

I. INTRODUCTION

Along with the rapidly development of the network, protocols have space distribution, simultaneity, asynchronous, instability and diversity. Designing a high quality protocol and protocol testing could never rely on intuition. Thus PE (Protocol Engineering) turns up. PE strictly uses formalization methods, technology and language in protocols design, protocol maintenance, protocol testing and various activities. So the formal description technique plays an important part in PE, which makes to get favorable safeguard for the correctness and reliability of the network protocol testing. The formal description technique becomes the focus which people pay more and more attention to.

Due to protocols is complicated, formal description languages are used to descript protocol specification in order to giving the clear and accurate meaning. Three types of formal description languages have been developed. They are ESTELL language, SDL language and LOTOS language. E-LOTOS(Enhancements to Language of Temporal Ordering Specification) is the definition of the revised version of the LOTOS standard [1]. E-LOTOS enriches LOTOS in many aspects, such as executable, more user friendly data types, time and exception handling, object-oriented etc. E-LOTOS with the solid theoretical foundation and the application technology may be applied to study systems with parallel, uncertain and synchronized concurrent. But the text E-LOTOS only descript static text model and cannot directly and effectively descript system structure and the levels and relationship among activities. And it is difficult to distinguish these expressions using the text LOTOS. The worst is that systems described by the text E-LOTOS cannot execute. All of these will restrict E-LOTOS. According to reference [2], interactive graphic model has been widely applied to formal description technology.

Reference [3] introduces currently two LOTOS graphic models, called WG-LOTOS and UO-LOTOS for LOTOS. WG-LOTOS not only uses more graphics styles, but also uses nested rectangle, which makes systems executable difficult. UO-LOTOS uses the hierarchical structure from top to bottom adopts the label like the text LOTOS. UO-LOTOS can express level structure of LOTOS. Whereas these graphic models are not widely used in especially industry, they only can express static structure and are not very flexible. Consequently, they are very difficult in execute, verification and testing. GE-LOTOS(Graphic E-LOTOS) is proposed in this paper. GE-LOTOS has the characteristics of visualization and intuitive, and supports logic detection, module design and testing etc.

II. E-LOTOS AND GE-LOTOS

A. E-LOTOS

E-LOTOS is a kind level of language. The top layer is the module language, while the bottom layer is the basic language. The module language of E-LOTOS can describe completely abstract syntax. This abstract syntax include: specification description, module, module expressions, general module description, interface, interface expression.

1. specification declaration:

   specification spec-name [import mod-exp1,...,mod-expn] is
   [gates G1:T1,...,Gm:Tm]
   [exceptions X1:T1,...,Xp:Tp]
   (behaviour B | value E )
   endspec

   spec-name is the name of specification; mod-expi expresses module expressions with the specification import; G1:T1,...,Gm:Tm expresses the gate parameter list; X1:T1,...,Xp:Tp expresses the exception parameter list; the body of specification is made up of behavior expressions or value expressions.

2. module declaration:

   The module is used to account for a series of relevant type, function and process. The module declaration in E-LOTOS is the following show:
   module mod-id [: int-exp] [import mod-exp1,...,mod-expn] is
m-body
endmod
mod-id is the name of module; int-exp is the interface expression. The module may be visible through the declaration of the interface expression. The definitions of various objects within the module are available; mod-expi expresses a series of module expression; m-body is made up of type definition, function definition and process definition.

B. GE-LOTOS Overview

GE-LOTOS uses protocol’s description of text E-LOTOS as input, and finally gives the GE-LOTOS presentation of protocol. GE-LOTOS has some incomparable advantages. GE-LOTOS is image and intuitive, and supports logic detection, module design and testing etc. The paper designs GE-LOTOS’ six basic graphic style and meaning. GE-LOTOS is a tree structure. The following form gives the corresponding E-LOTOS semantic of GE-LOTOS, see table 1.

Square: represents a operator, such as "||", "[", ",", ";", ">>", etc;

Rectangle: represents a event;

Diamond: represents a condition judgment;

Hexagon, represents a reserved word, like "for", "if", "case", "module", "function" etc;

Circle: represents a state;

Line: used for connecting all graphics.

Table 1 GE-LOTOS Basic Graphics Elements

<table>
<thead>
<tr>
<th>Text E-LOTOS Component</th>
<th>GE-LOTOS Presentation</th>
</tr>
</thead>
</table>
| Operators using square, such as: "||", "[", ",", ";", ">>", etc |  \[ \] \( \) \, \; \> \>
| Activities using rectangular, such as: stop | stop |
| Condition judgment using diamond, such as: ifB1 then | B1 |
| Keyword using hexagonal, such as: for, loop, if, case, spec, module, fun, proc, exc, p | for \( \), loop \( \), if \( \), case \( \), spec \( \), module \( \), fun \( \), proc \( \), exc \( \), p \( \) |
| State using circle, such as: Idle[propList: LinkList] | Idle[propList: LinkList] |

C. Conversion from E-LOTOS to GE-LOTOS

Conversion from E-LOTOS to GE-LOTOS to do the following aspects job:

Firstly, the text E-LOTOS will be processed. Before the input text E-LOTOS conversion to GE-LOTOS, the text LOTOS will be processed. The mainly reason is that the expression in the text E-LOTOS is infix form. Based on analyzing GE-LOTOS, it is displayed in the form of tree. Each of the operators is tree node. If the text E-LOTOS is prefix form, the drawings method will be simplified.

Secondly, Trees will be established for GE-LOTOS. GE-LOTOS is a longitudinal tree structure in the design using children-brother chain table for storing. When establishing the tree, only reading the prefix form after conversion may finish all trees. In GE-LOTOS, specification and module are two independent trees. Therefore, they will be established with respective trees.

Thirdly, GE-LOTOS will be drawn. Graphics draw is after establishing trees. When trees are processed by first root sequence traverse, it will draw the current visiting the GE-LOTOS node. When the node is being drawn, it is able to draw out the corresponding graphics through the content of judgment to the current node.

III. TTCN-3 ABSTRACT TEST SUITE OF PROTOCOL BASED ON GE-LOTOS

TTCN-3 (Testing and Test Control Notation 3) [4–6] is a language which specifically used to describe abstract test cases of OSI protocols’ conformance testing with characteristics of standardization and versatility. TTCN-3 is a text-based language. Its syntax and semantics is very similar to the common high-level programming languages. Its ability of the dynamic test configuration unifies a wide range of previous test architectures, so its descriptive power is enhanced greatly and the scale of application is also significantly extended. And the test system described with TTCN-3 can easily be transplanted to other test systems. The TTCN-3 top-level unit is a module, which is composed of the definition part and the control part. The part of the definition defines such as test components, Communication ports, data types, test data templates, characteristics of the process called on test ports and test cases, etc. The control part calls on the test cases and controls the execution of test cases[7].

Firstly, Generating TTCN-3 abstract test suite of protocol based on GE-LOTOS has need of the following information: the state set, the input event set, the output event set, and the input/output event set. Secondly, it also needs the leading sequence and the UIO sequence based on analyzing every state. Finally, the leading sequence, the UIO sequence, the input/output event, and combination of the type definition generate TTCN-3 abstract test suite.

Leading sequence: if the initial state is considered as FirstState, the sequence is made up of all input /output events from the FirstState to S state is the leading sequence of S state. According to the definition, the initial state based on GE-LOTOS may have multiple paths to reach the objective state. The leading sequence may be the one of these paths. Dijkstra algorithm can be used here for the shortest path sequence as the leading sequence of S state, namely, P (FirstState, S).
Test transition: the test transition is the sequence of events which the acceptable input event and the corresponding output events of S state constitute. Analyzing GE-LOTOS may get input/output event sequence, namely, Input/Output.

UIO sequence: the UIO sequence of S state is the test transition what S state can accept and other state can't accept, or the input/output events which is not acceptable other states.

The test case of S state is the combination of S state’s leading sequence, test transition and UIO sequence, namely, P(FirstState, S)&Input/Output&UIO(S). "&" represents the connection of string. Combined type conversion and test case can generate TTCN-3 abstract test suite of protocol. The type definition will be added into module. Each generating test case will be maked to subgroup and be added into module in form of TTCN-3 abstract test suite.

IV. GE-LOTOS REPRESENTATION AND TTCN-3 ABSTRACT TEST SUITE OF PROTOCOL BGP-4’S FINITE STATE MACHINE

Border gateway protocol BGP-4 is a kind of external gateway protocol, which realizes routing information of protocol exchange in different the autonomous system [8]. BGP-4 protocol state machine has six states, twenty-eight input events, twenty-one output events. The following example is GE-LOTOS representation and TTCN-3 abstract test suite of protocol BGP-4’s finite state machine. See figure 2 and figure 3.

Figure 2. BGP-4 module part's Connect state's GE-LOTOS representation

Figure 3. Test case of BGP-4 protocol test suite testing Connect state

V. CONCLUSIONS

Along with network playing more and more important role in society nowadays, formal description technology study will need more and more people. The paper realizes designing from the textual E-LOTOS to GE-LOTOS, and gives how generate TTCN-3 abstract test suite based on GE-LOTOS. Through the application for BGP-4, the paper gives the GE-LOTOS presentation and the TTCN-3 abstract test suite of the BGP-4’s finite state machine. And the paper lays a good foundation for further design generating executable test suite based on GE-LOTOS.

REFERENCES