

- System Adapter (SA), it can fit the communication between TTCN-3 testing system and the system under test by abstracting the testing system interfaces.
- Platform Adapter (PA), it makes TTCN-3 testing system to adapt to the specific executing platform, and realizes the external functions and the related methods to operate timer.
- Code/Decode (CD), it converts the exchange data between SA and the system under test, which makes them having the format to be handled by the receiver.
- Test Management (TM), it provides the user interface and manages the TTCN-3 testing system.
- Component Handling (CH), it is responsible for managing the distributed testing components.
- TTCN-3 Control Interface (TCI), it provides the standardized adaption mechanism facing the specific system under test for TM, CH and CD [4].

TA integrates SA, PA and CD. IT can complete the adaption among the testing systems, the system under test and the executing platform. Although, CD is included into TA, in fact it is just quoted by TA when executing, and it is implemented as separated entity. It is the core of development work. After the Abstract Test Suite (ATS) based on TTCN-3 core language is compiled by TTCN-3 compiler of TTworkbench, it becomes TE.

III. THE WEB APPLICATION TEST BASED ON TTCN-3

The Web application based on B/S is different from the traditional software. It is composed of mutual cooperation and independent software subsystems. The communication between the subsystems is done through the communication protocol, and the internal communication in a subsystem is done via messages [5]. Architecturally, the Web application software system is composed of multi-level software, including the client browser, the server application service subsystem and the database service subsystem.

Since TTCN-3 language has powerful functions, such as dynamic configuration, synchronous and asynchronous communication, etc. testers find it difficult in ensuring the consistency between the test design and the test implementation; and it also needs to have the auxiliary tools when facing the management and maintenance of the huge test systems. These problems will be paid more attention with the mature of TTCN-3 test language. Reverse analysis of the test systems based on TTCN-3 can help testers and maintainers control the system design from higher level, and test the consistency between the test design and the test implementation, which has important significance for test system maintenance, expansion and assessment.

In view of the Web application system, TE, TA and CD should complete the following functions according to the TTCN-3 testing system. The relations among them are shown as Fig. 3.

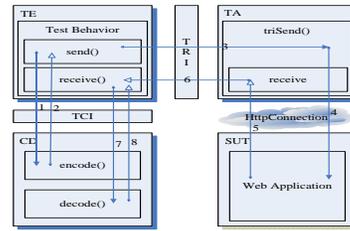


Figure 2. the flow chart of testing system execution

- TE, it defines test data and test behavior. The test data includes the excitation data input into the system under test and the response data output from the system under test.
- TA, it helps to implement the interactive behavior between the testing system and the system under test. Then it can input the excitation into the system.
- CD, it exchanges the coding data format, which makes the excitation data and the response data to be received by the system under test and the testing system.

The implementation process is as follows.

1. When the testing system executes the Send of TE, it firstly calls the Encode of CD in order to code the parameters from Send.
2. The incoming data (raw excitation data) will be coded by CD. The coded data will be back to TE.
3. The testing system sends the coded excitation data to the system under test by calling triSend of TA [6].
4. TA sends the excitation data to the system under test by using HTTP.
5. After the system under test deals with the excitation data, it transmits the response data back to TA by using HTTP. Because TA can receive the response data, it hasn't the method named Received.
6. TA transmits the raw response data back to TE.
7. The testing system decodes the raw response data to the format understood by it by calling the Decode of CD.
8. CD will send the decoded data back to TE aiming at being compared.

IV. THE IMPLEMENTATION SCHEME OF CODEC

The research shows that TTCN-3 is suitable for Web application testing, but there are also problems. Because all the pages of Web application are not same, it is necessary to design a codec scheme corresponding to each page. Therefore, in the development of TTCN-3 testing system for Web application, writing CD codes becomes the most complex and difficult work. However, in spite of different page contents, the structure characteristic of the page script is the same. So parsing the page script is general, which makes CD having domain generality. On this basis, providing a generic codec for Web application can improve the efficiency of the testing an development work, and can also enhance the applicability of TTCN-3 in the Web application testing.

As mentioned above, the SUT response that TA receives is a complete Web page to the TTCN-3 testing system. Its data format is HTML code, which implies the data that the

testers concern. CD parses the Web page according to the data definition of TE, and extracts the data, then converts its format. Different from testing the Web application with traditional C/S structure, the objects concerned by testers not only include data elements (that is the response results from the application to input data), but also include the page elements (that is the form of Web pages, such as the form, the text box etc.) in the course of testing the Web application with B/S. Therefore, the data which TE expects to receive is defined as a hierarchical structure, which can effectively describe the target data. The relevant research had been carried out by Pulei Xiong [7], shown as Fig. 4.

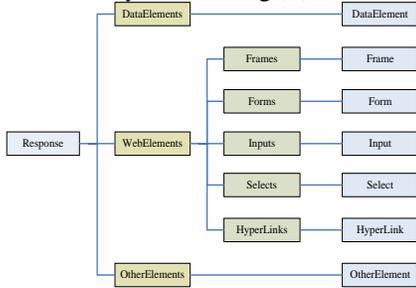


Figure 3. the expected data hierarchy

Because Frame, Form, Select, Input and Hyperlink are the common elements of Web pages, they are selected as WebElement, and other special or unique elements are classified as OtherElement. DataElement describes the specific data. It can improve the generality to define the expected receiving data by combining with WebElement, OtherElement and DataElement. The following code gives a specific definition.

```

module ResponseType {
  type set DataElement {
    charstring      DataField,
    charstring      DataValue
  }
  type set of DataElement DataElements;
  type set Frame {
    charstring      FrameName,
    charstring      FrameSrc
  }
  type set of Frame Frames;
  type set Form {
    charstring      FormName,
    charstring      FormAction,
    charstring      FormMethod
  }
  type set of Form Forms;
  type set Input {
    charstring      InputName,
    charstring      InputType
  }
  type set of Input Inputs;
  type set Select {
    charstring      SelectName,
    charstring      SelectId
  }
  optional,
  charstring      SelectClass
}

```

```

optional
}
type set of Select Selects;
type set Hyperlink {
  charstring      Href,
  charstring      Target optional
}
type set of Hyperlink Hyperlinks;
type set WebElements {
  Frames          frames,
  Forms           forms,
  Inputs          inputs,
  Selects         selects,
  Hyperlinks     hyperlinks
}
type set OtherElement {
  charstring      OEField,
  charstring      OEValue
}
type set of OtherElement OtherElements;
type set Response {
  DataElements   dataElements,
  WebElements    webElements,
  OtherElements  otherElements optional
}
}
}

```

In the above code, *module* is the top element of TTCN-3 code. Test data, configuration and behavior are defined in it. *Set* is an out-of-order structured type, which is similar to the *Struct* type of C. *Set of* is a collective description whose elements are *Set*. In the code, DataElement, Frame, Form, Input, Select, Hyperlink and OtherElement define seven basic elements of end nodes in Fig. 4. They contain the name/value pairs from respective attributes. DataElements, Frames, Forms, Inputs, Selects, Hyperlinks and Other Elements are the set of all the corresponding basic elements. WebElement is a set of Web elements. *Response* contains three sets of data elements, Web elements and other elements, which is corresponding to the entire page. The *Optional* of the code indicates that the item is optional, and may not exist in real data.

V. COVERSION OF TARGET DATA

This paper focuses on the generality which implies a certain degree of abstraction. CD is a complete unit of work which has specific input and processing corresponding to the input and output. Among these, the input is determined by the system under test, the corresponding output is different because the system under test is varying. From another perspective, although the input content of CD (that is original data) is complicated, it still has some similarities as a Web page, which determines the common codec should be reflected in the data processing method. Because the parsed Web page has complex structure, a large amount of information and interference information, the efficiency is very low if the received data is directly extracted from raw response data. According to the above characteristics, this paper adopts two-layer analytical strategy to decompose the problem. The target data will be obtained through two times

of continuous analysis, data format conversion will be completed by the assembly, shown as Fig. 5.

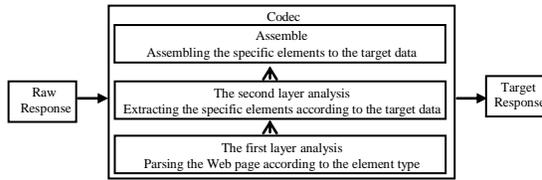


Figure 4. two-layer analytical strategy

The corresponding parsing rules are customized according to the different system under test, which can deal with the differently raw data. XML is a standard markup language, and has a standard method to describe the structured data. So the parsing rules are described by using XML, and the description will be input into each layer of parsing function module in the form of a document. The data flow of CD is represented as Fig. 6.

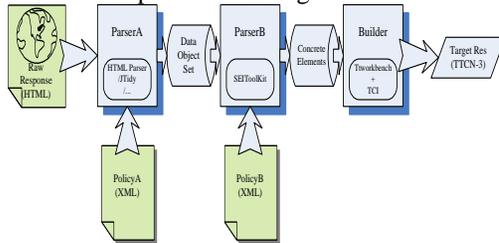


Figure 5. the flow chart of CD

Aiming at different systems under test, the analytical rule description file is only modified to parse the different targets by using the two-layer analyzing. The two-layer analyzing and assembling both implements the decode method of CD, The role of the first layer analysis is to filter irrelevant data, parse the raw data according to the data elements mentioned above, and form the data object sets corresponding to data element, Web page element and other elements. The first layer analysis is achieved by using the open source or library from the third party, such as HTML Parser, JTidy. The second layer analysis aims at the specific data elements contained in the target data from every data object set. Assembling aims at making the elements from the second layer analysis into the target data with the same data format as the desired receiving data, which can be done TCI interface that has function of the data conversion and data packaging. The expected receiving data defined in the above code is as an example. Seven data sets can be obtained after the first layer analysis, namely Data, Frame, Form, Input, Select, Hyperlink and other elements. The second layer extracts the specific elements in the desired receiving data which provides the material for assembly. The target data can be formed after assembly.

Of course, the design also exist some problems. Paring a Web page requires to input two analytical rule description files. If the system under test is a large scale of Web site, there will be a large number of the analytical rule description files. Therefore, it is necessary to abstract the

parsing rules in a certain degree, which can form the predefined parsing rules with model features. The predefined parsing rules are universal, which reduce the number of parsing rule description files and enhance the universality of encoder.

VI. SUMMARY

As a test standard, TTCN-3 is platform-independent and universal, which has been used in more and more fields, and is studied and extended both at home and abroad. From the TTCN-3 specification itself, both its core language and interface are the good foundations in every test application field. On the other hand, the TTCN-3 support tools provide the strong support for the spread of TTCN-3, such as TWorkbench based on Java and Exhaustif/TTCN based on C++. At present it is deficient in the relatively mature development methods of TTCN-3 testing system, this paper proposed a development method of testing Codec for Web application. Because of the large quantity of data and interference information of Web application, the raw data is processed by using the multilevel parsing, and then the target data will be acquired.

The purpose of the study is to propose a universal scheme for Web application software testing system based on TTCN-3, which helps to enhance the applicability of TTCN-3 specification in Web application software testing field. Therefore, the design of this method has been completed. The next step is to achieve the specific functions. In addition, this method is now focused on the data which is right or wrong, and the location distribution of Web page elements has not been considered, future research will focus on it.

ACKNOWLEDGMENT

This paper is contributed by National Key Laboratory of Software Development Environment Fund (SKLSDE-2010ZX-15) and High Technology Research and Development Program of China 863 Fund (2009AA01Z145).

REFERENCES

- [1] ETSI. European Standard (ES) 201 873-6 V3.1.1 (2005-06): Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 6: TTCN-3 Control Interface (TCI). European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, also published as ITU-T. Recommendation Z.140, 2005
- [2] ETSI. European Standard (ES) 201 873-5 V3.1.1 (2005-06): Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 5: TTCN-3 Runtime Interface (TRI). European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, also published as ITU-T. Recommendation Z.140, 2005
- [3] Robert L. Probert, Bernard Stepien, Pulei Xiong, Formal Testing of Web Content using TTCN-3, TTCN-3 User Conference 2005, Sophia-Antipolis, France, 2005
- [4] Lu Xiaoli. Research on Testing Technology of Web Applications [Degree Paper]. Northwest University, 2006 (In Chinese)
- [5] Ina Schieferdecker, Stephan Pietsch, Theofanis Vassiliou-Gioles, Systematic Testing of Internet Protocols- First Experiences in Using

TTCN-3 for SIP -, 5th Africom Conference on Communication Systems, Cape Town, South Africa, May 2001

- [6] Gao Xiang, Jiang Fan. A Survey of TTCN-3. Computer Engineering and Science, Vol 126, No 16, 2004 (In Chinese)
- [7] Liu Xiaoyong, Jiang Fan. Implementation of TTCN-3 Test System Based on Object-oriented Method. Computer Engineering, Vol 32, No 10, 2006 (In Chinese)