

XYZ/E and Its Application to the TIPS

Liu Man-xia¹, Zhang Jin¹⁺, LI Ye-long¹, Zhang Yu-xun¹, Liu Jun-fei²

¹ College of Information Science and Engineering, Hunan University, Changsha, China

² National Engineering Research Center for Software Engineering, Peking University, Beijing, China

Abstract. Treasury Information Process System (TIPS) is a large application system used by banks and tax department. The architecture design of TIPS is critical to its success. However, related research is rare. In this paper, a temporal logic language, XYZ/E, is used to describe the architecture of TIPS in order to prove rationality of TIPS's architecture. High-level architecture of TIPS is described from three different abstraction levels. The results show that the key components of TIPS and its structural relationship are revealed, which provide the foundation for the subsequent refinement and logic verification in the future.

Keywords: Formal Methods, XYZ/E, TIPS, Software Architecture

1. Introduction

With the rapid development of IT technology, the level of tax information management has become an important modernization symbol of tax administration. The tax information in United States is started to construct in 1960s. It aims to establish a comprehensive information management system to provide an efficient and fair way for taxpayers to pay tax. In China, TIPS is put forward by the People's Bank of China, the Ministry of Finance, and State Administration of Taxation in June, 2007. Declare inquiries on-line, deduct from account and duty-paid real-time is provided by TIPS. TIPS achieves to pay tax at home for taxpayer by solving the problems going back and forth between bank and tax department frequently.

TIPS provides tax declaration and payment through bank online for taxpayer. The research related to TIPS focus on the application of SOAP to bank tax network system [1] and the design and implementation of the mechanism of bank tax network system [2]. However, there are few studies on software architecture of TIPS, especially for system level design that the high-level organization and interaction between calculation units. The software architecture design is crucial to the development process of TIPS and the inappropriate software architecture will lead to disastrous consequences. The architecture description language is used to give out a formal description of software architecture, which provides the object for further research. XYZ/E temporal logic language is the representative formal description language designed by Prof. Tang Zhisong in Institute of Software, China Academy of Science [3]. XYZ/E based on linear time temporal logic proposed by Manna-Pnueli [4] is an effective software architecture description language. To express the state transition, only a few temporal operators in XYZ/E system are used. In addition to the first-order logic knowledge programmer familiar with, XYZ/E does not require users to have profound abstract mathematics training to understand and use XYZ/E [5]. A software architecture description language XYZ/ADL represented by graphical is established [6]. It achieves the gradual transition in different hierarchy architecture and combination between modularized program design method and stepwise refinement method. XYZ/E is applied to CA authentication system to describe the CA and RA component in formal method [7]. XYZ/E is applied to build hybrid real time systems of the steam boiler control specification problem [8]. A general compositional verification method in XYZ system concerning the sequential case is developed [9].

⁺ Corresponding author. Tel.: + (86)-731-88823294
E-mail address: jin2012zhang@gmail.com

First, this paper introduces the XYZ/E architecture description language. Second, client component, connector, server component of TIPS in XYZ/E are described. It reveals the relationship of the key components in TIPS by the description of semantic content and high level design in different hierarchy. It lays the foundation for further refinement and validation system logic.

2. XYZ/E and software architecture description

2.1. XYZ/E and software architecture

XYZ system is the combination of temporal logic language, software engineering methods and tools [10]. XYZ/E temporal logic language is the core of the XYZ system. It is not only a temporal logic system, but also a kind of programming language. The smooth transition of software system from elementary to advanced in the unified framework can be implemented in XYZ/E. The static and dynamic semantics is coordinated by introducing the concept of agent, which combines the static semantics of package and dynamic semantics of execution.

The software architecture is made up of component, connector and interactive segment. The component is a certain functional logic unit or a storage object. The component is divided into simple component and composite component. In order to describe software architecture, XYZ/E must be extended to support the explicitly description for the main unit in software architecture. After extending, the corresponding description of simple component, composite component and connector in XYZ/E as follows:

- Simple component.
`%PORT PortName= = DataType Declaration; [PortBehavior]`
`%FUNCTION= = [Function Specification]`
`%COMPUTATION= = [ComputationSpecification]`
- Composite component
`%COMPOSITION== [ComponentInstanceName:ComponentName; ...`
`ConnectorInstanceName:ConnectorName; ...]`
- Connector
`%ATTACHMENTS== [ComponentInstanceName.PortName#ConnectorInstanceName.RoleName; ...`
`ComponentInstanceName. PortName ## PortName; ...]`

2.2. Description on TIPS

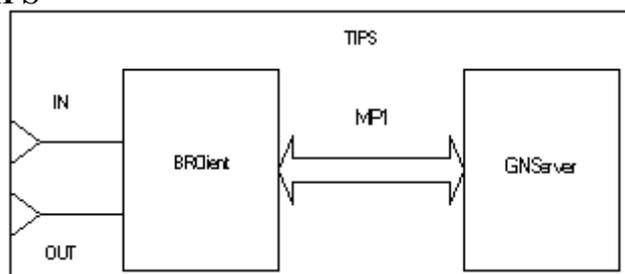


Fig. 1: Component model of TIPS

The component model of TIPS is shown as Figure 1. TIPS can be seen as a composite component. It is made up with BRClient component, GNServer component, MPI connector. BRClient component communicates with GNServer component through MPI connector. TIPS component marked with TIPS is in the highest level of TIPS. Users interact with system through the IN port and OUT port. GNServer component is responsible for responding to request from BRClient component. According to TCP/IP protocol, GNServer and BRClient component initiate communication requests and transmit data to each other.

The TIPS component is described first, then the internal components BRClient component, MPI connector, and GNServer component. TIPS component is in the highest level of abstraction in software architecture. The description of TIPS component in XYZ/E is shown as follows:

```

%SYSTEM TIPS= = [
%GLOB  Request:Message; Result:Message           //the global variable
%PORT  IN= = MESSAGE;                             //an instance of the message delivery
        □[LB=START-IN=>IN?Request^$O LB=y1]      //user input his request
%PORT  OUT= =MESSAGE;
        □[LB=y2=>OUT! Result^$OLB=y3]
%PROPERTY= = [Properties]
%FUNCTION= = [Request→◇Result]                   // functional specification;
%COMPOSITION= = [BRClient: client; GNServer: server; MP1: mp]
%ATTACHMENTS= = [BRClient. Input ## IN; BRClient. Output ## OUT] // bind ports
%COMPUTATION= = □ [LB= START-IN=> $O LB= y0;
                  LB=y0=>$OIN= BRClient.Input^$OOUT=BRClient.Output^$OLB=y1;
                  LB=y1=>[[ BRClient.computation; MP1.GLUE; GNServer. computation];
                  LB= End=> $O LB= EXIT]]

```

The client is mainly distributed in various tax rooms. Taxpayers just log in and choose a deductible item, then click the OK button. The deductible request will be transmitted to the server through the connector.

The description of the BRClient component in XYZ/E is shown as follows:

```

%COMPONENT BRClient= = [
%PORT Input= =MESSAGE;
        □[ LB= START-IN=> $O LB= y0;
          LB= L0^ Input? logininformation=> $O LB= L1;
          LB=L1=>Input? Request^$O LB=L2]
%PORT Output= = MESSAGE;
        □[ LB= L7=> Output! Result^ $O LB= EXIT]
%FUNCTION= = [ input→◇ output]
%COMPUTATION= =
        □[ LB= START-IN=> $O LB= L0;
          LB= L0^Input? logininformation=> $O LB=L1;
          LB= L1=> MP1. sender (logininformation)^ $O LB= L2;
          LB= L2 => MP1. receiver (loginresult)^ $O LB= L3;
          LB= L3^~ (Isloginresult= $T)=> $O (LB= L4| EXIT);
          LB= L4=> Output! (relogin)^ $O LB= L0; //check fails, Re login
          LB= L5=> Input? Request^$O LB= L6;
          LB= L6=> Protocol (Request) ^$O LB= L7;
          LB= L7=> MP1. sender (Request) ^$O LB= L8;

```

```

LB= L8=> MP1. receiver(Result) ^ $O LB= L9;
LB= L9=> Output! Result ^ $O LB= END]

```

]

Upon receiving the request, the server will authenticate identify, and take different actions according to different roles. The roles are divided into three, taxpayer, administrator and super administrator. There is a backup of user information in the tax department in advance and the user doesn't need to register. After processing the request, the server sends the results to the client. If failed to log in, it will ask user to login in again.

The description of the GNServer component in XYZ/E is shown as follows:

```

%COMPONENT GNServer == [
%FUNCTION= = [process→◇ result] //function specification
%COMPUTATION= =□[ LB= START-PR=> $O LB= P0;
    LB= P0=> MP1. receiver (logininformation)=> $O LB= P1;
    LB= P1=> process (logininformation)^ $O LB= NEXT;
    LB= P2=> Protocol (Result, loginresult)^ $O LB= NEXT; // verification
    LB= P3=> MP1. sender (loginresult)^ $O LB= EXIT;
    LB= P4^MP1. receiver (Request) => $O LB= P5;
    ? [ LB= P5^ Request. user= TaxPayer=> $O LB= P51
        | Request. user= Administrator=> $O LB= P52
        | Request. user= Super Administrator=> $O LB= P53;
    LB= P51^ process1 (Request)=> $O LB= EXIT
    LB= P52^ process2 (Request)=> $O LB= EXIT
    LB= P53^ process3 (Request)=> $O LB= EXIT] //operations for roles
    LB= P6=> MP1. sender (Result)^ $O LB= END]
]

```

The server is responsible for processing the deductible request from the client. The function is completed in Computation sub-component.

The description of the Computation component in XYZ/E is shown as follows:

```

%COMPONENT GNServer.COMPUTATION= = [
%PORT Input= MESSAGE;
    □[LB= START- Input => Input ?Request^ $O LB= z1]
%PORT Output=MESSAGE;
    □[LB= START- Output => Result ?Request^ $O LB= EXIT]
% COMPUTATION= =
    □[LB= START-PR=> $O LB= z0;
        LB= z1^ Input? Request. DeductionTax= $O LB= z2;
        LB= z2^~ (AccountBalance< PayingTax)=> $O (LB= z3| EXIT);
        LB= z3=> $O IsPaid= True$O LB= z4;
    ]
]

```

LB= z4=> \$O Output! deductsuccess/\ \$O LB= z5;
LB= z5=> \$O LB= EXIT]

]

3. Conclusion

Because the research on TIPS lacks of software architecture in domestic, in the view of software architecture, the client component, server component and connector in TIPS are described in XYZ/E from different levels of abstraction. It reveals the important semantic content and high level design of TIPS and promotes the effective software development. XYZ/E as a formal method and UML 2.0 as a graphic method are combined with to provide more intuitive and accurate description of TIPS in the future.

4. Acknowledgements

This paper is supported by National Natural Science Foundation (60901080) and China Postdoctoral Science Foundation (20100480219).

5. References

- [1] J. X. Liu, X. Yang, and H. H. Li. Application of SOAP in tax-bank on net project. *Computer Engineer.* 2003, **29** (15): 173-175. [\(In Chinese\)](#)
- [2] Z. Q. Gao. Design and implementation of the network system among finance bureau taxation, national treasury and bank based on security mechanism. *Computer System & and Application.* 2010, **19** (8): 128-132. [\(In Chinese\)](#)
- [3] Z. S. Tang. Logic Programming and Software Engineering (Volume2). Science Press, 2002. [\(In Chinese\)](#)
- [4] Z. Manna, A. Pnueli. The Temporal Logic of Reactive and Concurrent System: Specification. New York: Springer-Verlag, 1992.
- [5] Z. S. Tang. Logic Programming and Software Engineering (Volume1). Science Press, 1999. [\(In Chinese\)](#)
- [6] G. Q. Zhang, J. D. Zheng, and J. H. Luo. An architecture description language based on temporal logic its visual support environment. *Computer Engineering and Applications.* 2001, **37** (5): 14-16. [\(In Chinese\)](#)
- [7] J. Y. Liu, G. Q. Zhang, M. Rong. The description and refinement of CA attestation system based on XYZ/E. *Computer Science.* 2006, **33** (8): 267-270. [\(In Chinese\)](#)
- [8] A. Yan, Z. S. Tang. Building Hybrid Real Time Systems in XYZ/ E Implementation of the Steam Boiler Control Specification Problem. *Journal of Software.* 2000, **11** (6): 711-719.
- [9] H. L. Xie. A Structured Temporal Logic Language: XYZ/ SE. *Journal of Computer Science and Technology.* 1991, **6** (1): 1-10.
- [10] Z. S. Tang. The goal, meaning, effect and application of the XYZ system. *Journal of Software.* 1999, **10** (4): 337-341. [\(In Chinese\)](#)