Research on the Reconfiguration Router Unit Component Composition Technology Based on the Agent

Liu Qiang
National Digital Switching System Research Center, Zhengzhou , China
Email: lq_strong@163.com

Wang Binqiang
National Digital Switching System Research Center, Zhengzhou , China
Email: wbq@mail.ndsc.com.cn

Huang Wanwei
National Digital Switching System Research Center, Zhengzhou , China
Email: huangww79@163.com

Jiang Nan
PLA general logistic department Archives department, Beijing, China
Email: baobao_0312@yahoo.com.cn

***Abstract*—Component based reconfiguration router unit (RRU) is one of the most effective solutions to improve function extended. It can satisfy the urgent need of multi-network and multi-business. Component composition and compositional reasoning are the core technologies and frontier research areas in RRU. Based on the characteristics of RRU and software components and inspired by software component reusing, 3 component composition mechanisms are proposed in this paper to integrate software components simply and conveniently. And it is argued to compose interfaces at the same time of component composition, consequently to generate more powerful and more abstract interfaces to support integration of coarse-grained components and raise the abstract level of component composition. Moreover, based on the Wright's research on formal specification of software architecture, compositional reasoning algorithms about the behaviors of composite component as well as the protocols of composite interfaces are developed, which establish a foundation to analyze, validate, simulate composite systems. Meanwhile, according to the characters of RRU, we also consider the limited of resource in component composition process. At last we present the implement of component composition.**

***Index Terms*—Router, reconfiguration, component composition, agent, resource limited**

I. INTRODUCTION

As function extended has become a hotspot in router research field[1][2][3][4], the traditional architecture has been challenged. The traditional router has close architecture, and each network unit is made by one manufacturer. The manufacturer owns all the technologies, including software and hardware. User only can control the equipment via the software supplied by the manufacturer, so the reconfiguration, such as programmable and user-oriented setup is limited is limited. To break the monopoly of router produce and meet the integration need of multi-network and multi-business, we designed reconfigurable router unit. The design idea is to bring in component based network equipment design technology, which will turn the equipment manufacturer into suppler of component development. The network operators can choose or customize business processor according to special needs and construct according network equipment into architecture in a building block component composition, which can effectively shorten the development period. In the component design, components from different manufacturer can interconnection through unified component standard, so the previous close network will become open. As each component can be update, added and deleted, the component based router owns reconfiguration.

Component composition and composition reasoning mechanism are two key technologies in the reconfiguration router unit[5][6]. Component composition mechanism is constricting software system via multi-component. Compositional reasoning is predicting the fuctionability and property of software system according to the fuctionability and property of components.

Based on the characteristics of components, this paper describes the component relationship, in the

This work was supported in part by the China 863 high-tecnology plan (2008AA01A323)

Liu qiang is with the National Digital Switching System Research Center, Zhengzhou , 450002, China (phone: 86-371-81632946; e-mail: lq_strong@163.cn).

doi:10.4304/jcp.5.2.218-225

reconfiguration router unit software architecture from down to up, and proposes the formal semantics of the reconfiguration router unit software model, which can improve the component design quality, opening, programming and high-reconfiguration.

Through the analysis of current technology and based on the consideration of router design characteristics, the following core problems need to be solved in the process of component composition and compositional reasoning:

1) Regulate the software component description of the reconfiguration router unit and classify the methods the component composition.

2) Solve the service component of the reconfiguration router unit, generate more powerful and more abstract interfaces though component composition and formulate of the reconfiguration router unit architectures.

3) When the required the reconfiguration router unit component is absence or unusable, how to field suitable the reconfiguration router unit component to replace it under the condition of non-interruption of the whole flow.

Based on the characteristics of the reconfiguration router unit and inspired by the process construction method in process algebra, this paper proposes 3 RRU component composition mechanisms, which can integrate software components flexibly and simply. Deepening the thoughts of plug and socket architecture[7][8][9], this paper argues to compose interfaces at the same time of component consequently to generate more powerful and more abstract interfaces to support integration of coarse-grained components and raise the abstract level of component composition. Moreover, to realize the auto-composition of the reconfiguration router unit components, formal specification of service composition is proposed, which can realize the composition flow via software agent. The agent is a self-control function entity, which can evaluate and adjust its behavior according to the sense of environment.

Meanwhile, since the specification of the reconfiguration router unit function, each component is composed by communication serial; this paper gives the reasoning algorithms combined with the component composition mechanisms. Section 2,3 gives the software description of the reconfiguration router unit and classifies the component composition methods. Section 4 gives the requirment of resource in the component composition process.Section 5,6 gives the composition scheme based on agent and implement. Section 7 concludes the paper and points out the future work.

II. FORMAL DESCRIPTION OF THE RECONFIGURATION ROUTER UNIT SOFTWARE COMPONENT

A. the definition of software component

Software component is the key element of the reconfiguration router unit software plane, and the physical entity for system function. So each functional modular which processes on the software plane can see as a RRU component. In router switch equipment, software component can be a logic unit which can deal with simple packets or a multi-function modular which can process complex packets, when standard interface protocol is made. Every two components which have the same interface can communicate. Compound component is defined as top component which is composed by some components and has capacity to implement complex functions contrariety, function unit used to compose compound component is basic component on different abstract layer the concept of basic component and compound component are relative. Integrated compound component can be used as a basic element for other compound component on high layer, in which condition it is seen as basic component on the higher layer.

From the view of system integration, component is an encapsulated functional unit, and has the black-box characteristic, so its communication with other components must through its interface. In router switch equipment, the Potential to deal with the object for the software component is input packet packets. Form the view of component composition the core part is packet flow among components, which means input packets go in through which input interface and output packets go out through which output interface after processing. As software component is event triggered, the trigger condition must be set in the component. The input object only can be processed when it satisfies special trigger mechanism. In other words, the input packets of a component must satisfy its input constraint to get output result; otherwise the software component will be in empty running state.

Based on above characteristic, software component is defined as a quintuple integrated by input interface set, trigger, event set, function set and output interface set. It can be written as SC (software Component) (S, I, C, F, O)[10][11] and its black-box module is shown in figure 1, S represents the basic description information of this component, such as manufacturer, version, size and so on. F represents the basic function set of this component, which describe the functions can be processed by the component, including the process of data flow and trigger of each trigger state. I represents the input interface set of this component, such as shows this component has three input interfaces. O represents the output interface set of this component. C represents the trigger condition set of this component, which means only the process object X, which satisfies the trigger condition will be processed.

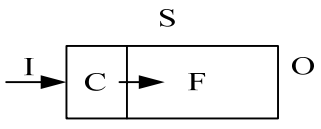

Figure 1. the definition of quintuple model of basic component

Let's see the component which supports IPv4 packets transfer process. Since the interfaces among components designed based of standard, components can communicate with each other, and this component can support input packets with other types, but only IPv4

packets be processed by his component, the packets of other types will be discarded. So the trigger condition of the component, which supports IPv4 packets is $C = \{IPv4\}$. This component includes many complicated process flow, such as key word extraction and table look up, but these details are packaged by interfaces, and formulated by F. The description of F directly shows the granularity of the component.

B. the defination of component composition

On the RRU switch platform, the process object of software component is input package. Based on the property of software component, the formal description of process can describe the process flow of the package[12]. The formal description of process flow simply describes the package operation at component interface, and veiled the inner complicated process. The formal description based on propriety are described as blow

$$\forall x \to I_1, \exists y \to O_1 \mid x \in C, y = F(x, y) \quad (1)$$

$$\forall x \to I_1, \exists \varnothing \to O_1 \mid x \notin C \quad (2)$$

In formula 1, $\forall x \to I_1$ shows input object depends on input interface I_1, which means input packet goes through the I_1 interface of this component. When the input package satisfies the trigger condition of this component, $x \in C(x)$,the input package will be processed and send out its process result to output interface O1. It means output result y depends on output interface O1, and can be shown as $y \to O_1$, formula 2 shows that if the input package doesn't satisfy the trigger condition of this component, the input package won't be processed. In the software components process flow, the simplest function is to connect the output interface to the input interface directly. The formal description is $\forall x \to I_1, \exists y \to O_1 \mid C = \varnothing, y = x$, $C = \varnothing$ shows none trigger condition is needed, and $y = x$ shows the input interface and output interface connect directly.

Here is the example of component which supports IPv4 package transfer. This component if described as SC_forward (In x : I ; Constrain IPv4: C; Process forward_IPv4 : F; Out y : O; forward IPv4 packages : S). It can be see from its formal property definition that this component has an input interface and an output interface, and it supports the package of IPv4 type. The formal description of the inner process flow of this component is $\forall x \to I, \exists y \to O \mid x \in \{IPv4\}, y = forward\_IPv4(x, y)$.

III. COMPONENT COMPOSITION ANALYSIS

The plane development based on software components is a process of multi-component composition. The thoughts is similar to the plug and socket architecture in software project. It's realized by gradually improving the abstract level of complex component. The basic components which communication with each other and have closed relationships are integrated to compound composition. So the inner complicated processes can be veiled. The compound component is in favor for component composition with bigger granularity, which can save the scale of the system composed of components. With the abstract level of high component composition is raised, the software platform with be abstracted to the highest level compound component.

In the process of component composition, different components need to communicate with each other, and the communication style decides the transmission road of processed object. Based on the flow direction of processed object, there are two basic communication styles. One is unsplitted, which is called as atom-composition. It has three process flows serial, offshoot and aggregation. The three composition styles said above can make any compound components. Based on the formal description of basic components, the formal description of compound components made from the three styles are proposed. Then based on the formal description of compound components, the components of higher level are proposed, in the end, the language description style of the whole complicated software system can be proposed.

A. serial composition

On the router switch platform, packets go through different components in flow line style, so packets are commonly processed in serial style. In the serial composition, SC1 and SC2 which respectively supports ahead and back process are in series. As shown is figure 2, the formal description of two components are SC1 and SC2. As defined by the quintuple integrated in this module, the input of component SC1 can be processes only if it satisfies its trigger condition. The input of SC2 is subset of SC1, and it also needs to satisfy the trigger condition of component SC2. In serial composition, through putting the output of SC2 to the input of SC1, the process object can be circulately processed, so the circulation process flow is a special example of serial flow.

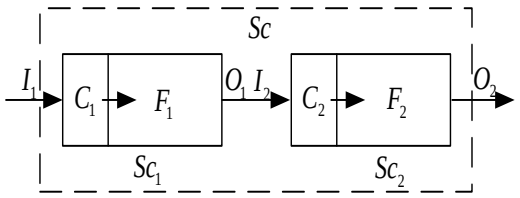

Figure 2. serial compound component composition

$$\forall x \to I_1, \exists y \in D \mid x \in C_1, y = F_1(x, y), y \to O_1 \quad (4)$$

$$\forall y \to I_2, z \to O_2 \mid y \in D, y \in C_2, z = F_2(y, z) \quad (5)$$

Formula 4 shows the input of component SC1 must satisfy its trigger condition to get process result. The output of SC1 is the middle variable of compound component, and middle set variable D is used to describe the possible output result of SC1. Formula 5 shows the input of component SC2 is the subset of the output of SC1, which is also the subset of middle set variable D,

the input of SC2 must satisfy its trigger condition to get output result.

As said above, SC= SC1+ SC2 , the symbol" +" shows the two components use serial process flow. The interfaces of SC are input interface I1and output interface O2. In the SC, the function of component SC1 is processed firstly, then the function of component SC2. The quintuple integrated of compound component SC is SC(S,I,C,F,O), I= I_1 ,C={ $C_1(x)$, $C_2(y)$ }, F={ $F_1(x,y)+F_2(y,z)$ },O= O_2 .

B. offshoot composition

Offshoot composition means the two basic components of one compound component sharing one input interface. The input of this interface can selecting is sent to one of the two basic components. After be splitted at the input interface, the input will be sent to two indepent components to be processed. The basic component SC1 and SC2 in an offshoot composition is encapsulated in one compound component SC1 as shown is figure 3.

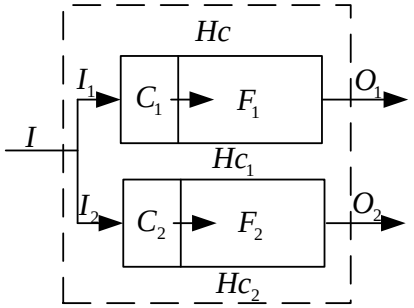

Figure 3. offshoot compound component composition

$$\forall x \to I, y \to O_1 \mid I_1 = I, x \in C_1, y = F_1(x,y) \quad (8)$$

$$\forall z \to I, w \to O_2 \mid I_2 = I, z \in C_2, w = F_2(z,w) \quad (9)$$

Formula 8 shows the input of component SC1 goes through interface I. It must satisfy the trigger condition C1 and the process result goes through interface O1. Formula 9 shows the input of the component SC2 also goes through interface I. It must satisfy the trigger condition C2 and the process result goes through interface O2.

As said above, the component SC=SC $\vee$ SC , and the symbol " $\vee$ " describes the selective relationship between these two software components. The interfaces of this compound component are input interface { I } and output interface { O_1 , O_2 }. The function of SC1 and SC2 can be simultaneously processed inside. The quintuple integrated of compound component SC (S,I,C,F,O), I= I ,C={ $C_1(x)$, $C_2(z)$ },F= $F_1(x,y) \wedge F_2(z,w)$, O={ O_1 , O_2 }.

In offshoot composition, the relationships of SC1 and SC2 are parellel, copy, selection and interruption.

Definition 1. (component parellel composition) The parellel composition of component P and Q is P||Q. P||Q is a new component. The compound interface protocol of P||Q is the interface protocol of P and Q parellel processed. The function of P||Q is the function of P and Q parellel processed. A compound component with higher function can be gotten through parellel composition, which can supply the function of those two components simultaneously.

Definition 2. (Component selection composition) the selective composition of P and Q is P□ Q, P□ Q is a new component. The compound interface protocol of P□ Q is processing the interface protocol of P or Q according to outside environment. The function of P□ Q is processing the function of P or Q according to outside environment. A compound component with higher function can be gotten through selection composition, which can supply the function of these two components, but the two functions can't be processed simultaneously.

Definition 3.(component copy composition)the copy composition of P is P@n(n ∈ N). P@n is a new component. The compound interface protocol of P@n is parellel processing n interface protocols of P. The function of P@n is parellel processing n functions of P. A compound component with higher function can be gotten through copy composition, which can parellel supply the many functions of one component simultaneously. It can improve performance and reliability.

Definition 4.(component interruption composition) the interruption composition of P and Q is P ∧ Q. P ∧ Q is a new component. The compound interface protocol of P Q is processing the interface protocol of P firstly, but stopping it as soon as the interface protocol of Q is started. The function of P ∧ Q is processing the function of P firstly, but shopping it as soon as the function of Q is started. Interruption composition is used in exception handles and recovery.

C. aggregation composition

Aggregation composition means the two basic components in one compound component shave one output interface. The two components have respective process object, and their process results aggregated to one output interface. In aggregation composition, SC1 and SC2 are encapsulated in one compound component SC1 as show in figure 4.

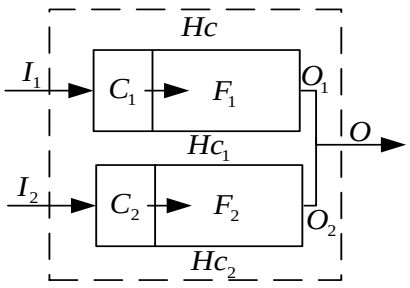

Figure 4. aggregation compound component composition

$$\forall x \to I_1, y \to O \mid x \in C_1, y = F_1(x,y), O = O_1 \quad (10)$$

$$\forall z \to I_2, w \to O \mid z \in C_2, w = F_2(z,w), O = O_2 \quad (11)$$

Formula 10 shows the input of component SC1 goes through I1. It satisfies trigger condition C1 and will be processed. The process result will be aggregated to output interface O. Formula 11shows the input of the component

SC2 also goes through interface I. It must satisfy the trigger condition C2 and the process result goes through interface O2.

As said above, compound component SC= SC1 ∨ SC2 , and the symbol ' ∨ ' means the aggregation process of the output results from these two components. The interfaces of compound component are input interface $\{I_1, I_2\}$ and output interface $\{O\}$. The functions of SC1 and SC2 are independently processed at the same time. The quintuple integrated of SC is SC(S,I,C,F,O),I=$\{I_1, I_2\}$,C=$\{C_1(x), C_2(z)\}$,F=$F_1(x,y) \vee F_2(z,w)$,O=O .

As in above, the relationships of SC1 and SC2 also can be parted into four composition condition: parellel, copy, selection and interruption.

Ⅳ. THE REQUIREMENT OF RESOURCE

As the result of the particularity of the RRU, whether the process of composition satisfy the restrict of resource should be think over in the process of composition[13-21].

1) By the restrict of the resources in the process of component composition system, whether all of the manner.
2) Check out whether the function manner in the system of component composition satisfy the restrict of resource. By the restrict of resource, whether the function we concerned in the composition system satisfy the restrict of resource.

In the paper, we give some basic suppose of the using of resource.

1) The using of resource is monopolizing. When the resource was in use, the other component can not use it. The resource can not be shared.
2) The resource can be reclaimed, but we didn't think over the cost of the reclaiming at present.

Definition 5. Suppose the process of resource using is $P(S_p,A_p,R_p)$,S_p is the finite state concourse of component,A_p is the finite action concourse of component, R_p is the resource restrict of components. The process is controlled by the agent. $P_{S_0} \xrightarrow[R_0?]{A_0} P_{S_1} \xrightarrow[R_1?]{A_1} P_{S_2} \ldots P_{S_{m-1}} \xrightarrow[R_{m-1}?]{A_{m-1}} P_{S_m}$

We can see from the definition 5, P expression in the components interface behavior change state process, the components use the environmental resources dynamics. Component J in the 0 state and request of the Agent of resources R, to meet the requirements for A1 for component port changes in the composition or movement.

Current-path saves the components composition traverse path in the process of component composition, B is the length of storage of the resource constraints for the vector k; we use X (SC) marks the current visit to SC resources component vector, abnormal focus on the process of storing the traversal to find the abnormal component, satisfied is a Boolean variable, when the traversal process when there is no abnormal elements, satisfied for the "true" otherwise "false".

Algorithm 1 Resources Test Algorithm
Current_path={SC_0},B=<b1,b2,b3…bn>
X=<0>,abnormal=∅ satisfied=true
Do
{
Find the next component
SC=get the last component in current_path
If SC components have been the follow-up visited **then** delete the last component
Else begin SC get the follow-up component that SC did not take a visit to the follow-up component
X=X(SC)
For i=0 to K-1
If X[i]>bi then satisfied=true
If satisfied=true **then** SC join to abnormal Collection;
Join SC to current_path;
End
}
While(current_path={})
If abnormal= **then return** true **else return** false

Ⅴ. COMPONENT COMPOSITION SCHEME AND DEDUCE

The monitor and maintain of the reconfiguration router unit composition flow is realizes by software agent. The input supplied by flow maker must be transformed to the behavior standard set of agent through parser, which will be used to diver agent to monitor flows. Agent gets messages of other components in the reconfiguration router unit through a group of agent clients in RRU component. These messages will be transformed into component object which can be recognized by agent build-in rule engine, then the behavior regulate said above can be triggered. The ability of an agent includes definition a behavior set previously. The behavior set includes using some RRU component through effectors simulation, updating the inner variables via received messages and selecting an execution from a candidate set based on user scheme. After a behavior is triggered, the execution will be managed by a behavior execution engine[22][23].

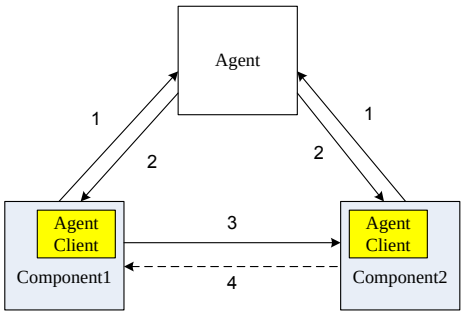

Figure 5. Agent-based components interactive

the main function of the Agent:

a) Management the registration, cancellation and information storage of component, the realization of components can be dynamically loaded, unloaded and Component Retrieval;

b) Can send commands to the components, the realization of components to uninstall or re-starting;

c) The communication of components (IP address and port number socket) notice of the other components connected to achieve the support of distributed systems;

d) Through heartbeat signals sent to the component elements of Inquiry, and the maintenance of the various components connected topology.

The main function of the Agent Client:

a) Encapsulate the external interface of Component

b) Provide the function of communication between the Agent and component and component and component; realize a wide range of inter-process communication, so that components do not have to care about the details of communications; for shielding components specific communication details of the underlying operating system to achieve the support of heterogeneous systems.

Process of component interactions

Figure 5 shows, Agent and Component1, Component2 can be deployed in local or distributed systems, interactive process between the two components are as follows:

a) system initialization, Agent read the configuration file, configuration file keeps the whole system component connections; Agent checks whether it is reasonable to connect components; Component1 and initialization Component2 registered to the Agent to inform the Agent's own means of communication and address; Agent in accordance with configuration files and the various components of the registration information to determine the interrelationship between the various component connections.

b) Agent will be registered and address of the communications transmitted to the components associated with (mainly to give the message of the other component elements, that is, the upstream component).

c) When the Component2 need to request the service of Component1, from the Agent's address to send information directly to the Component2;

d) Component2 received news, the implementation of the operation and when circumstances Component1 Send feedback message to inform the implementation of the results.

e)After registration, Agent and Agent Client heartbeat signal between the Agent for the realization of the monitoring component, when the elements cease to function, Agent responsible for notifying the other components associated with (the upstream component), and update the database component connections.

In component composition, the main goal is to bind component interfaces through composition, which will turn simple component interface into compound component interface. Because in software development, components are managed through processes, we introduce Wright's CSP to analyses those style: selection, serial, interruption, copy and parellel.

The port m of example of the internal components N bound to the port complex components k, or to participate in the assembly complex component k, then there is a corresponding port binding, assembly transform function S, the three-tier structure computation process of N transformation events n.m.o into k.o, said that because the port binding or composition, the event o of port n.m as the event o of the outside pork k.Suppose a compound component has V interface used for port binding and composition exchange, and these functions combined as S.

The port m of example of the internal components N participate in the assembly of an external port k, there are examples of a corresponding transform port function T, the three-tier structure for the incident n.m.e transform the structure of an event e, because in the derivation process, the composite interface interaction agreement process with the three-tier structure of the incident, the need for symbolic transformation, with a structure with atomic CSP component in the process of port incident as an assembly to achieve transparent, can not be determined from the internal component is a compound component or basic components.

Suppose V ports binded to port K or composed to port K, the composition of their corresponded instance exchange function is described as T.

1) the deduce of selective composition serial composition and interruption composition. if OP is the symbol of composition scheme OP ∈ { □ ,;, ∧ },Q1,Q2,…,Qn, Compound component COM composed of n components, and the computation process of COM is S(OP i:1…n:Qi) suppose m inner component ports Q1.P1,Q2.P2,…,Qm.Pm, are binded encapsulated to the port K of the compound component or composed as port K, the behavior communication process of this port is (OP i:1…m.Qi). If the composition scheme is selection, serial and interruption, the computation process of the compound component is each inner component processes package through selection, serial and interruption execution. The port binding and composition can be gotten through symbol countercharge via function S. The protocol process of compound interface is composed of relative inner port processes which connect through selective, serial or interruption computation symbol.

2) The behavior deduce of copy composition and parellel composition copy composition is a special example of parellel composition, so it has uniform composition deduce scheme as parellel composition. Suppose compound component COM is composed of n components Q1,Q2,…,Qn in parellel style. The computation process of COM is S(||i:1…n.Q:). suppose m ports Q1.P1,Q2.P2,…,Qm.Pm, are binded to port K, or composed as port K, the process of port K is T(||i:1…Qi.Pi). If the composition scheme is copy and parellel, the computation process of the compound component is each inner component process package parellel. The port binding and composition can be gotten through symbol count change via function S. The protocol process of compound interface is parellel composed of relative inner interface protocol. But the parellel composition scheme of CSP orders that same events from two process letter tables must be extended simultaneously. The three level counterchange of

(||i:1…m.Qi.Pi) will be converted into one level constructor, and the communication protocol process of this compound interface can be gotten.

Ⅵ.IMPLEMENT

Since the characteristic of the reconfiguration router unit especially it will run in many network environments, the function of a compound component must be composed of many single component. For example, there is Ethernet interface component, ATM interface component and PoS interface component, which respectively forward relative package through Ethernet interface. ATM interface and PoS interface, an interface component of the reconfiguration router unit can be composed of these three interfaces, which can receive IP package, PPPoE package and ATM package.

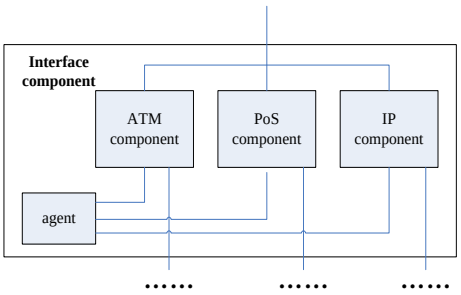

Figure 6. Agent-based interface component composition

In component composition, one inner interfaces are bund to the three outside interface of compound component which is used as input component of IP network. The 3 interfaces first send resource required to the agent, and agent calculate the resource spending and transfer the composition system to composite the three basic component into a compound component. When this compound is used to compose with other components, the three interfaces, Ethernet interface, ATM interface and PoS interface must be differentiated and mastered, so they can be selectively used depend on different conditions. Interface composition is realized in the process of component composition. Ethernet interface, ATM interface, and PoS interface are composed into one interface which can receive package of three different types and process the package head relatively.

Ⅶ.CONCLUSION

Component composition and composition reasoning are the core of the reconfiguration router unit. This paper proposes three component composition schemes based on the software component characteristic and process composition in process algebra. Any complicated software system can be composed via these three component composition scheme. Deepening the thoughts of plug and socket architecture, this paper proposes interface composition at the same time of component composition. The relative inner interface can be organized into an outside interface, and encapsulated supply services of more complicated and abstract function. This can support component composition with bigger granularity and improve its abstract level. Meanwhile, based on the CSP formal regular software architecture is proposed by Wright. This paper gives the deduce algorithm of the function behavior and communication protocol of such compound component. This deduces algorithm also improves the quality of compound software system and furthers the goals of predictable component composition of the reconfiguration router unit. The further work includes non-function property deduce, and more instance modeling and analysis.

ACKNOWLEDGMENT

The authors would like to thank the editor and the anonymous reviews for the comment. This work was supported in part by a grant from “863 high-tech plan” (2008AA01A323).

REFERENCES

[1] Decasper, D., et. al., “Router Plugins", Washington University Tech Report WUCS-98-08, February 1998
[2] Robert Morris, Eddie Kohler, John Jannotti, and M. Frans Kaashoek. The Click modular router. ACM Transactions on Computer Systems, Vol.18, No.3, August 2000, Pages 263–297.
[3] S. Karlin, L. Peterson, VERA: An Extensible Router Architecture, in: Proceedings of. the 4th International Conference on Open Architectures and Network Programming.(OPENARCH), 2001, pp. 3–14.
[4] Houidi, I., Louati, W., Zeghlache, D., "An extensible software router data-path for dynamic low-level service deployment", IEEE Workshop on High Performance Switching and Routing 2006, Poland, June 2006, pp. 161-166.
[5] Yang FQ,Mei H,Li KQ.Software reuse and software component technology.Acta Elecetronica Sinca,1999,27(2):68-75
[6] Zhang SK,Zhang WJ,Chang X,Wang LF,Yang FQ.Building and assembling reusable components based on software architecturre.Journal of Software, 2001,12(9):1351-1359
[7] Luckham D,Vera J,Meldal S.Three concepts of system architecture.Technical Report.,CSL-TR-95-674,Stanford University,1995
[8] Hoare CAR.Communicating Sequential Processes.Prentice Hall,1985
[9] Canal C,Fuentes L,Pimentel E,Torya jm,Vallecillo A.Extending CORBA interfaces with protocols.The Computer Journal,2001,44(5):448-462
[10] Murali S.Compositional performance resoning.In Harris CC,ed.Proceedings of ICSE CBSE4.IEEE,2001.98-101
[11] Nenad M,Richard NT.A classification and comparison framework for software architecture description languages.IEEE transactions on software engineering,2000,26(1):70-93
[12] Houidi, I., Louati, W., Zeghlache, D., "An extensible software router data-path for dynamic low-level service deployment", IEEE Workshop on High Performance Switching and Routing 2006, Poland, June 2006, pp. 161-166.

[13] Yang FQ.Thinking on the development of software engineering technology.Journal of Software,2005,16(1):1-7(in Chinese with English abstract). http://www.jos.org.cn/1000-9825/16/1.htm
[14] Lu J,Ma XX,Tao XP,Xu F,Hu H.Research and progress on Internetware.Science in China(Series E),2006,36(10):1037-1080(in Chinese with Englishabstract)
[15] De Alfaro L,Henzinger TA.Interface theories for component-based design. In: Henzinger TA,Kirsch CM,eds.Proc of the EMSOFT 2001,LNCS 2211,Springer-Verlag,2001
[16] De Alfaro L,Henzinger TA.Interface automata.In Proc.of theJoint 8th European Software Engineering Conf. and 9th ACM SIGSOFT Int'l Symp. On the foundations of software engineering(ESEC/FSE 2001).ACM Press,2001.109-120
[17] Burmester S,Gehrke M,Giese H,OberthAur S.Making mechatronic agents resource-aware in oder to enable safe dynamic resource allocation.In:Buttazzo GC,ed.Proc. of the 4th ACM Int'l Conf. on Embeded software,EMSOFT 2004.ACM,2004.175-183
[18] Miki-Rakic M,Medvidovic N.Architecture-Level support for software component deployment in resource constrained environments. In:Bishop JM,ed.Component Deployment,IFIP/ACM Working Conf.,CD 2002.LNCS 2370,Springer-Verlag,2002.31-50
[19] Fredriksson J,SandstrAom K,Akerholm M.Optimizing resource usage in componnet based real-time systems.In:Heineman GT,Crnkovic I,Schmidt HW,Stafford JA,Szypeiski KC,eds.Component-based software Engineering, the 8th Int'l Symp.,CBSE2005. LNCS 3489,Springer-Verlag,2005.49-65
[20] Fredriksson J,Tivoli M,Crnkovic I.A component-based development framewoek for supporting function and non-functional analysis in control system design.In:Redmiles DF,Ellman T,Zisman A,eds.Proc.of the 20th IEEE/ACM Int'l Conf. of Aumated software Engineering.ACM Press,2005.368-371
[21] Mousavi M,Reniers M,Basten T,Chaudron M.PARS:A process algebra with resources and schedules.In:Larseen KG,Niebert P,eds.Proc. of the Int'l Conf. on Formal Modeling and Analysis of Times Systems. LNCS2791, Springer-Verlag, 2004.130-150
[22] Martin RC. Agile Software Development: Principles , Patterns, and Practices. New York: Prentice Hall,2002
[23] Jia Y. The evolutionary component-based software reuse approach[PhD Thesis]Columbus The Ohio State University,1995

Liu Qiang received B.S., M.S. in National Digital Switching System Research Center in 2004, 2007. Now liuqiang is the Ph.D candidacy in National Digital Switching System Research Center. His interests include next network technology, software reconfiguration technology, reconfiguration router unit.

His job is research on the software technology of the reconfiguration router unit.

Huang WanWei received B.S.,M.S. and Ph.D. degress in National Digital Switching System Research Center in 2002, 2005 and 2008. His interests include FPGA technology reconfigure computer and reconfigure router.

Wang binqiang received B.S in Nanjing Aeronautics and Astronautics University in 1982,and received M.S. in Northwest Telecommunications Engineering collage, and received Ph.D in Xidian University. Since 1997, he has been with National Digital Switching System Research Center. His interests include next network technology, reconfiguration router unit.

Jiang Nan received B.S in Nanjing politcs academy in 2005, and received M.S. in Logistic command academy in 2008. Her interests include next network technology, software reconfiguration technology, reconfiguration router unit.