形式化与可视化相结合的软件体系结构描述方法研究*)

戎 玫[1] 张广泉[2,3]

(暨南大学深圳旅游学院 深圳518053)[1] (苏州大学计算机科学与技术学院 苏州215006)[2]

(重庆师范大学计算机科学学院 重庆400047)[3]

摘 要 软件体系结构是软件工程领域中一个重要的研究内容,研究软件体系结构的首要问题是如何描述一个软件系统的体系结构模型。本文通过集成XYZ/ADL与UML两种描述方法在软件体系结构中的应用,寻求一种基于时序逻辑理论的形式化方法与面向对象的可视化方法相结合的软件体系结构描述新途径。着重研究XYZ/ADL与UML在电梯控制系统体系结构建模中的应用问题,并运用基于构件的求精方法对该系统的主要组件进行了求精。

关键词 软件体系结构,建模,形式化方法,可视化方法,XYZ/ADL,UML

Software Architecture Description Approach Integrating Formal Methods and Visual Methods

RONG Mei[1] ZHANG Guang-Quan[2,3]

(ShenZhen Tourism College, Jinan University, Shenzhen518053)[1]

(Computer Science and Technology School, Suzhou University, Suzhou215006)[2]

(Computer Science School, Chongqing Normal University, Chongqing400047)[3]

Abstract Software architecture is an immature field in the discipline of software engineering. The first issue of the research is how to represent and describe the software architecture models of a system. The main purpose of this paper is to find out the effective approaches of integrating formal methods and visual methods in modeling software architecture by attempting the theory of SA into practical development. The effective approaches of integrating UML and XYZ/ADL are emphasized during modeling the software architecture of Elevator Control System. The major components are refined according to the component-based refinement method.

Keywords Software architecture, Modeling, Formal methods, Visual methods, XYZ/ADL, UML

1 引言

软件体系结构(Software Architecture)是近年来软件工程领域一个重要的研究内容[1]。与传统软件技术研究的不同之处在于,软件体系结构把原来从不同角度和不同观点出发建立的分散、非系统、非形式化的关于软件构成的思想经过研究分析和归纳综合,建立统一可用于软件系统设计、分析、构造、实现的符号和概念体系。根据1999年IEEE AWG/P1471的定义,所谓软件体系结构,它是软件系统的高层抽象,描述整个系统的结构和行为模型,明确了主要的系统组件(component)、组件之间的交互——连接件(connector)以及组件和连接件如何结合在一起的各种约束条件与可见属性[2]。对于大型复杂软件系统的开发,其体系结构设计的重要性已远远超过特定算法和数据结构的选择,成为决定系统成败的关键因素。

研究软件体系结构的首要问题是如何表示和描述体系结构。目前主要有两类描述方法:一是以体系结构描述语言ADL为代表的形式化描述方法,二是以统一建模语言UML为代表的可视化描述方法。ADL通常以一形式化语言作为它的底层语义模型,并为软件系统的概念体系结构建模提供具体语法和概念框架,还有一系列基于底层语义的工具支持体系结构的表示、分析、演化和设计;ADL不但是形式化描述软件体系结构的基本工具,而且也是对软件体系结构进行求精、验证、演化和分析的前提和基础;目前国外常见的ADL有Wright、Rapide、Unicon、Aesop、Darwin、ACME、SADL…等。文[3,4]提出一种基于可执行时序逻辑语言XYZ/E的软件体系结构描述语言XYZ/ADL,并采用XYZ/ADL描述了体系结构的组件、连接件等基本元素以及几种重要的体系结构风格[5]。但是,ADL不足之处在于其抽象程度高,采用了专用符号,难以被开发人员所理解,不便于交流和使用,较难融入到当前软件开发的实践中。

UML是一种将软件开发过程中出现的各种模型用可视化图形来描述的语言。它融合了多种面向对象开发方法的优点,采用统一的图形和符号从多个视角描述软件系统的各种抽象模型,获得了国际标准化组织的认可,并被国际软件界广泛接纳。但是,作为一种通用语言,UML易于理解和交流,其缺点是对软件体系结构的可构造性建模能力较弱,缺乏形式化语义,对体系结构的描述只能到达非形式化的层次,不利于系统的求精和验证[6]。

综上所述,ADL与UML两种描述语言各有特色,比较它们各自的优缺点,我们可以看出,ADL与UML在描述软件体系结构方面具有很强的互补性。ADL形式化语义的精确性正好可以弥补UML非形式化的一些不足,二者的有机结合,不仅可更精确地描述体系结构模型,而且还能支持下一步的求精和验证。基于此,本文以一个典型的实时系统——电梯控制系统为例,初步探讨UML和XYZ/ADL相结合来描述

*)本项研究得到国家自然科学基金(60073020);中国科学院计算机科学国家重点实验室开放课题(SYSKF 0303);重庆市教委科学技术研究项目(040803);江苏省计算机信息处理技术重点实验室开放课题基金资助.戎 玫 副教授,博士,主要研究方向为软件工程、人工智能等。

软件体系结构模型，并运用基于构件的求精方法对该系统的主要组件进行了求精。

2 基于时序逻辑的软件体系结构描述语言 XYZ/ADL

首先简要介绍一下基于时序逻辑的体系结构描述语言 XYZ/ADL 与基于构件的软件体系结构求精方法。

2.1 XYZ/ADL 简介

XYZ/ADL 是对时序逻辑语言 XYZ/E 的扩充。XYZ/E 是中科院唐稚松院士提出的世界上第一个可执行的时序逻辑语言[3]，XYZ/ADL 描述软件体系结构的单元主要有组件、连接件和交互端[4]。其中组件又可分为复合组件和简单组件。

(1)简单组件：

```
%PORT PortName==DataType Declaration;[PortBehavior]
%FUNCTION==[Function Specification]
%COMPUTATION==[Computation Specification]
```

(2)连接件：

```
%ROLE RoleName==DataType Declaration;[RoleBehavior]
%GLUE==[Interact Protocol]
```

(3)复合组件：

```
%COMPOSITION==[ ComponentInstanceName:Component-
    Name;…
    ConnectorInstanceName:ConnectorName;… ]
%ATTACHMENTS==
    ComponentInstanceName. PortName # ConnectorIns-
    tanceName. RoleName;…
    ComponentInstanceName. PortName # # PortName;… ]
```

运用 XYZ/ADL 描述软件体系结构模型，就是对体系结构进行抽象描述并逐步求精的过程。限于篇幅，关于 XYZ/ADL 的详细内容见文[3,4]。

2.2 基于构件的体系结构求精方法概述

文[2]介绍了软件体系结构的三种求精方法：基于行为替代的体系结构求精、基于风格的体系结构求精和基于构件的体系结构求精。基于构件的体系结构求精过程可以演化为两个过程：从粗粒度构件向细粒度构件的求精；从非形式化到形式化的求精。其基本思路是：确定了风格的体系结构级粗粒度构件按照规范说明，划分成多个构件，构件又可以按照需求确定自身的子风格，逐步形成细粒度构件。下文我们将采用基于构件的体系结构求精方法来建立电梯系统体系结构的模型。

3 实时控制系统建模与求精[1]

目前，实时控制系统在国民经济和国防现代化建设中的地位愈来愈重要，其研究具有重要意义，已成为计算机科学与控制领域的一个重点研究内容[7]。有关实时系统的建模方法已有多种，如 CORBA 方法、Peri 网方法等[7]。下面以一个典型的实时系统——电梯控制系统为例，初步探讨 UML 和 XYZ/ADL 相结合来描述软件体系结构模型与求精问题。

3.1 电梯控制系统简介

电梯控制系统是一个典型的实时控制系统，我们以某20层的商办中心的5部电梯组(标识为 A-E)的电梯系统为开发背景。电梯系统可分为三个部分：输入输出设备、群控调度系统和五部电梯各自的控制系统及装置。由于系统的高度对称性和面向对象的特性及组件复用的特性，可把对整个群控系统的建模简化为对单台电梯控制系统的建模[8]，故我们仅建立一部电梯控制系统的模型。

(1)输入输出设备：用户输入信号的按钮面板，包括每个楼层的召唤(外召)和轿厢召唤(内召)；以及系统向用户显示运行状态的发声或发光等显示设备。

(2)群控调度系统：主要与楼层按钮、轿厢控制器(5个)通信。它在接收到呼梯信号及电梯状态信息后进行计算，将呼梯信号分配给实际电梯系统中的一部电梯。分派的结果将随着群控器的调度算法的不同而不同。

(3)电梯的控制系统：从性质上可分为逻辑控制系统和拖动控制系统两部分，电梯的正常运行过程是通过电梯的逻辑控制系统与拖动控制系统之间紧密配合完成的。

(4)电梯系统还有井道、轿厢、门系统和应急装置等附属设备。此外，电梯系统还有许多其它附属装置(如自动平层装置等)，限于篇幅，在此不作一一介绍。

3.2 电梯控制系统的需求模型

在电梯系统中，我们采用 UML 用例图来描述系统的需求模型(见图1)，它用来描述人及外部设备和系统的相互作用[8]。电梯系统与外界的交互及提供的功能主要表现为：电梯系统运送乘客到目的楼层；而分布于井道的传感器则为电梯系统提供电梯的位置信息，间接作用于电梯的运行及故障判断中；分布于门系统的传感器向系统传送门状态信息；电梯拖动系统接受逻辑控制系统的指令，并依靠机械设备拖动电梯上下运行。所以电梯控制系统的主要使用者是：乘客(Passenger)、传感器(Sensor)和拖动装置(Dragcontroller)。系统提供以下用例：处理乘客输入信号：包括来自楼层的呼梯信号和来自轿厢的召唤信号；显示电梯状态包括被调度电梯序号、方向、轿厢所在楼层、载重、到达目的楼层；载客运行包括：电梯上/下行、开关门、有故障时触动紧急开关等。

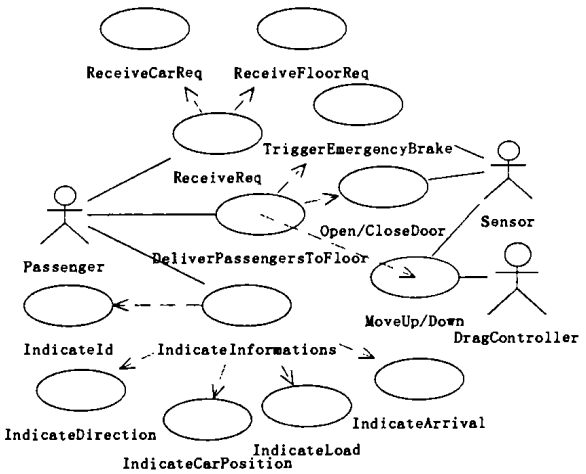

图1 电梯系统的需求模型(UML 用例图)

3.3 电梯控制系统的框架模型

框架模型侧重于描述系统的总体结构，在较粗的粒度上反映了电梯系统组件之间的关联，如图2所示。整个群控电梯系统总体上可划分为楼层组件(Floor)、电梯系统组件(ElevatorSys)、门系统组件(Door)和电梯群控调度系统(Elevator-Group-Dispatcher)四类复合组件。楼层组件主要包括呼梯按钮(Button)、按钮响应灯(Backlight)、电梯状态显示器(FloorIndicator)，此外，每个楼层上还有一个楼层到达传感器，用以检测电梯是否到达该楼层。单台电梯系统组件(ElevatorSys)主要包括电梯显示设备(ElevatorIndicator)、按钮(Button)及按钮灯(Backlight)和电梯控制系统，电梯控制系统是由电梯调度控制系统(ElevatorDispatcher)和拖动系统

1 张玲红硕士参加了本文实例系统的建模工作

(DragCo-ntroller)组成的。每部电梯和每层楼都有一扇门，故可抽象出父类组件 DoorOwner。门系统组件包括压力传感器(PressureSensor)、位置传感器(PositionSensor)和发动机(DCMotor)三个子组件。

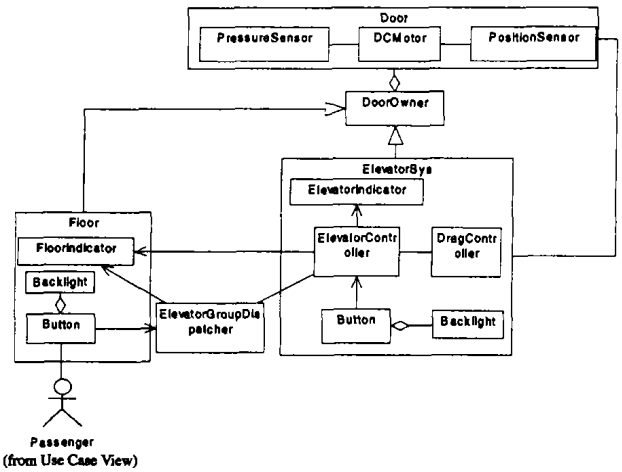

图2 电梯系统的框架模型(UML 类图)

3.4 电梯控制系统的体系结构风格

我们从使用者与系统的交互角度来分析框架模型，电梯系统的体系结构可视为一种特殊的控模块-视图-控制器(MVC)风格[1]：楼层和电梯的召唤按钮组件为控制器，接受用户的请求；显示设备组件相当于视图，向用户显示电梯运行状态；而电梯系统的主体部分，包括电梯控制系统、群控调度系统、电梯、门系统等都可视为模型中的组件，用以处理并响应用户输入的请求，并更新显示设备的信息。所以系统的组件包括按钮、电梯控制系统和显示设备，而连接件为它们之间的通信和交互。我们运用 XYZ/ADL 描述 MVC 这样一种风格，如图3所示。我们预定义了端口传输数据类型、端口类型和角色类型，并对连接件和组件所必须满足的约束条件进行了描述。

```
// MVC 体系结构风格描述
  %STYLE MVC==[
      %TYPE MESSAGE
        //定义端口传输数据的类型 MESSAGE
      %PORTTYPE In==MESSAGE //定义端口类型
      %PORTTYPE Out==MESSAGE
      %CONNECTOR MP //定义连接件
      %ROLETYPE Sink==MESSAGE //定义角色类型
      %ROLETYPE Source==MESSAGE
      WHERE[ $ Acon:Connectors(Type(con)=MP)//定义约束部分
          ∧ $ Acon:Connectors、$ Ar:Roles(con)(Type(r)=Sink $ ∨Type(r)=Source)
          ∧ $ Acom:Components、$ Ap:Ports(com)(Type(p)=Send $ ∨Type(p)=Receive)]
      //电梯系统体系结构风格的描述
        %System ElevatorSys==[
          %GLOB req:MESSAGE;state:MESSAGE
          %PORT In==MESSAGE;
                  □[LB = Start_call ⇒ In? req ∧ $OLB=r1]
          %PORT Out==MESSAGE;
                  □[LB = r2 => Out! state ∧ $OLB=r3]
          %FUNCTION==[req→◇state] //系统总体功能规范：系统对乘客呼叫进行响应
          %COMPOSITION ==[Button:Controller; //电梯系统的主要组件及连接件
                  Gelevatorsys:Modle;
                  Indicator:View;
                  Cbg:MP;
                  Cgi:MP]
          %ATTATCHMENTS ==[Button. Send # Cbg. Source;//组件与连接件之间的交互
                  Gelevatorsys. Receive #Cbg. Sink;
                  Gelevatorsys. Send #Cgi. Source;
                  Indicatorsys. Receive #Cgi. Sink;
                  Button. Receive # # In; //复合组件与简单组件的端口绑定
                  Indicator. Out # # Out]
          %COMPUTATION==[LB=Start_call=> $OLB=r1;
                  LB = r1 => $OIn = Button. Receive ∧ $OOut=Indicator. Out ∧ $OLB=r2;
                  LB = r2 => ‖[Button. COMPUTATION; Cbg. GLUE; Gelevatorsys. COMPUTATION; Cgi. GLUE; Indicator. COMPUTATION];
                  LB=End=> $OLB=EXIT]]
```

图3 电梯系统体系结构风格的描述

3.5 电梯控制系统的结构模型

电梯系统有很多信号，使得井道线路十分复杂。如何处理这些传输信号，对电梯控制系统极为重要。下面建立电梯控制系统结构模型。电梯逻辑控制系统实时接收来自井道、厅内外召唤、轿厢、门、楼层和群控的信号，并将这些信号按一定的逻辑关系进行转换并进行综合处理，将其处理结果如电梯运行方向、开关门控制、运行状态和运行速度等信号返回给原呼叫组件，以控制电梯的运行。图4对逻辑控制组件的第一步求精，它反映了主要组件之间的信号传输关系。

```
%COMPONENT Elevatorsys==[
  %PORT Receive1==MESSAGE;
        □[LB=START_receive=>Receive1?carreq ∧ $OLB=e1
  %PORT Receive2==MESSAGE;□[LB=START_receive=>Receive2?command ∧ $OLB=e1]//接收群控组件命令
  %PORT Send1==MESSAGE;
      □[LB=e0=>Send1!state1 ∧ $OLB=e1] //信号传向View 组件
  %PORT Send2==MESSAGE;
      □[LB=e0=>Send2!state2 ∧ $OLB=e1] //信号传向群控调度系统组件
  %COMPOSITION ==[Logicalcontroller; Dragcontroller; Safecontroller; Buildset; Doorcontroller; Elevator;
                  Cld:MP; Cdl:Mp; Cls:MP; Csl:MP;
                  Cbl:MP; Cldo:MP; Cdol:MP]
  %ATTATCHMENTS ==[Logicalcontroller. Send3 # Cld. Source;
                  Dragcontroller. Receive1 # Cld. Sink;
                  ……
                  Logicalcontroller. Send2 # # Send2] %COMPUTATION ==[LB = START_receive1 => $OLB=e1;
  LB=e1=>$OReceive1=Logicalcontroller. Receive1 ∧ $OReceive2=Logicalcontroller. Receive2
        ∧ $OSend1 = Logicalcontroller. Send1 ∧ $OSend2 = Logicalcontroller. Send2 ∧ $OLB=e2;
  LB=e2=>‖[Logicalcontroller. COMPUTATION; Cld. GLUE; Cdl. GLUE; Dragcontroller. COMPUTATION; Cls. GLUE;
  Csl. GLUE; Safecontroller. COMPUTATION; Clb. GLUE; Cbl. GLUE; Buildset. COMPUTATION; Cldo. GLUE;
  Cdol. GLUE; Doorcontroller. COMP-UTATION]]
```

图4 电梯系统体系结构的组件结构模型

3.6 电梯控制系统的动态模型

一般说来，电梯至少有3种运行状态：即正常运行、消防运行、检修运行[8]。其中，检修运行有最高级别，只有撤消检修运行，电梯才能转入其它状态。其次是消防运行，在消防开关动作后，电梯应立即返回底层或基站，当消防开关复原后，电梯应能立即转入正常运行状态。电梯在正常运行中，又可分为三种状态：①空闲状态(idle)：此时电梯停止在某一楼层，且电梯没有呼梯信号，包括目的层信号和呼梯信号；②等待状态(waiting)：此时电梯停在某一楼层，但电梯已有呼梯信号要

进行响应，或此时乘客正在进出电梯；③运行状态(motion)：电梯正在响应某个呼梯信号，可分为向上运行、向下运行或停止三种行为。

电梯运行的直接驱动力是乘客产生的呼梯召唤信号，这些信号又可分为上行信号和下行信号。要建立正确的电梯模型，必须了解电梯的运行规则，在运行状态中，电梯对信号的响应遵循以下原则：①顺向载车原则：电梯一旦按确定方向运行，只响应同向呼梯层信号，减速停车，记忆反向呼梯信号，等换向后再响应。但在满载时直驶不响应梯层信号，只响应目的层信号。②最远程反向载车原则：电梯向上运行时，对于向下的呼梯层信号，电梯先响应最远的，换向后，再按顺向载车的原则响应向下方向的其它信号。反之亦然。③自动开门原则：电梯到达某层后自动开门，延时2～10秒后自动关门。但电梯超载不关门并报警。据此建立电梯运行的动态模型如图5所示，这也是对电梯逻辑控制系统组件的求精。

电梯在运行中对呼梯信号的响应是一种请求应答方式，电梯必须在接收到电梯控制器返回的控制指令后才能确定下一步如何动作，求精过程如图6所示。这也是对电梯组件的求精，体系结构求精到这一层，基本上就是可执行程序了。

```
%COMPONENT Logicalcontroller==[
    %PORT Receive1==MESSAGE;
    □[LB=START_receive1=>Receive1?carreq∧ $OLB=r1]
    ……
    %PORT Send5==MESSAGE;
    □[LB=e0=>Send5!codoor∧ $OLB=s5]
    %COMPUTATION==[LB=START=>initiate∧ $OLB=
    y0;//初始化电梯系统
    LB=y0∧Receive5?buildinfo.carpos=> $OLB=y1;//接收到
    来自井道的轿厢位置信号
    LB=y1=>Send1!carstate1.currentfloor∧ $OLB=y2;//发送
    当前楼层信息
    LB=y2∧Receive3?carstate=> $OLB=y4;//发送轿厢状态
    信息
    LB=y4∧～(carstate=MOTION)=> $O(LB=y5|LB=
    y51);//判断轿厢是否为运动状态
    LB=y5∧(Receive1?carreq∧Receive2?groupcommand)=>
    $OLB=y6;//非运动状态时接收呼梯信号
    LB=y51∧Receive5?doorinfo∧=> $OLB=y52;//接收到门
    机信号
    LB=y52∧delay(tdo)∧process(doorinfo)=> $OLB=y53;//
    延时并处理门信号
    LB=y53∧～(tdo>6s)=> $O(LB=y52|LB=y54);//延时
    超过6秒关门
    LB=y54=>Send5!codoor(c)∧ $OLB=y6;
    LB=y6∧～(carreq=EOF)=> $O(LB=y8|LB=y7);//若无
    内呼信号，则处理群控指令
    LB=y7∧～(groupcommand=EOF)=> $O(LB=y8|LB=
    y4);
    LB=y8∧～(carstate1.direction=UP $∨carcarstate1.direction=DOWN)=> $O(LB=y9|LB=y81);
    LB=y81∧～(carreq=EOF)=> $O(LB=y811|LB=y812);
    LB=y812=>del(carstate1.direction)∧ $OLB=y9;//若有内
    呼，删除原方向
    LB=y811=>carstate.direction=carcarstate1.direction∧
    $OLB=y10;//若无内呼，则为原方向
    LB=y9=>decide(carstate.direction)∧ $OLB=y10;//确定
    方向
    LB=y10=>Send3!command.start∧ $OLB=y11;//发送运
    动指令
    LB=y11∧Receive3?carstate∧ $OLB=y12;
    LB=y12=>‖[Send1!carstate1;Send2!carstate2]∧ $OLB=
    y13
    LB=y13∧～(Receive5?buildinfo.decvel)=> $O(LB=y11|
    LB=y14);//接收来自井道的减速与平层信号
    LB=y14=>Send3!command.decvel∧ $OLB=y15;//发送运
    动指令减速信号
    LB=y15=>∧～(Receive5?buildinfo.arrflo)=> $O(LB=
    y15|LB=y2)]]//接收平层信号，电梯到达
```

图5　电梯逻辑控制系统组件的求精

```
%COMPONENT Elevator==[
    %PORT Receive1==MESSAGE;
    □[LB=START_receive1=>Receive1?command∧ $OLB=
    re1]
    %PORT Send1==MESSAGE;
    □[LB=START_send1=>Send1!carstate=> $OLB=se1]
    %PORT Send2==MESSAGE;
    □[LB=START_send2=>Send2!Loadwarning=> $OLB=
    se2]
    %COMPUTATION==[
        LB=START∧Receive1?command=> $OLB=zs;//接
        收到指令
        LB=zs∧～(command=EOF)=> $O(LB=z0|LB=
        start);//判断指令是否为空
        LB=z0=>initiate∧Turnonpower∧ $OLB=z1;//初始
        化
        LB=z1∧～(Isoverweight= $T)=> $O(LB=z2|
        z11);//判断是否超载
        LB=z11=>Send2!Loadwarning=> $OLB=z12;
        LB=z12∧Process(weight) $W～(Isoverweight= $T)=
        >LB=z2;
        LB=z2∧Receive1?command.move=> $OLB=z3;//接
        收到运动指令
        LB=z3∧Process(move)=> $OLB=z4;
        LB=z4∧Receive1?command.decvel∧ $OLB=z5;//接
        收到减速信号
        LB=z5∧～(command.decvel=EOF)=> $O(LB=z6|
        z4);
        LB=z6∧motion.state=decvel $U((Receive1?com-
        mand.arrflo)//减速运行直到平层
        ∧～(command.arrflo=EOF))=> $OLB=z8;
        LB=z8=>Send1!carstate=> $OLB=END]]
```

图6　电梯组件的求精

结束语　目前，已有一些关于ADL与UML相结合的工作，它们一般是通过UML的扩展机制集成和扩展ADL语义[9]、或将一些形式化工具(如CSP等)融入UML机制中[10]来描述体系结构模型。本文通过集成XYZ/ADL与UML两种描述方法在软件体系结构中的应用，寻求一种基于时序逻辑理论的形式化方法与面向对象的可视化方法相结合的软件体系结构描述新途径。通过一个实例，运用UML各种视图从不同侧面描述电梯控制系统体系结构模型，相当于一个个离散的对象，而运用XYZ/ADL对电梯控制系统组件逐步求精则实现了这些对象之间平滑的衔接，并通过连接件角色与组件端口的交互机制以及复合组件的端口绑定机制最终把系统形成为一个有序组合的模板。由于电梯控制系统是一个相对复杂的实时系统，本文侧重描述了系统的静态模型和各主要组件之间的交互模型，这也是形式化方法与可视化方法相结合在实时系统开发应用中的一个初步尝试。

参考文献

1　Shaw M, Garlan D. Software Architecture: perspectives on an emerging discipline. America: Prentic Hall, 1996

2　戎玫，张广泉. 软件体系结构求精方法研究. 计算机科学，2003，30(4)：108～110

3　唐稚松，等. 时序逻辑程序设计与软件工程. 北京：科学出版社，2002

4　朱雪阳，唐稚松. 基于时序逻辑的软件体系结构描述语言XYZ/ADL. 软件学报，2003，14(4)：713～720

5　Zhang Guangquan. Describing Software Architecture Styles with XYZ/E. In: Proc. of the 5th Inter. Confer. for Young Comp. Sci. International Academic Pblisher, ICYCS-2003 2003, 8: 171～174

6　张玲红，戎玫，张广泉. UML在运输业务管理系统建模中的应用. 计算机工程与应用，2004，40(14)：207～209

7　Gomaa H. 姜昊，周靖译. 并发与实时系统软件设计. 北京：清华大学出版社，2003

8　刘晓君. 电梯运行状态的规范与级别. 中国电梯，2003，14(4)：42～43

9　Guelfi N, Perrouin G. Rigourous Engineering of Software Architectures: Integrating ADLs, UML and Development Methodologies: [Rapport Technique TR-DIA-02-08]. Luxembourg University of Applied Sciences, 2002. 12

10　于卫，蔡希尧. 软件体系结构的描述方法研究. 计算机研究与发展，2000，38(10)：1185～1191