

基于 UML 的 SDH 网管系统的分析与设计

孙知信 王汝传 王绍楦

(南京邮电学院计算机科学与技术系 南京210003)

摘 要 网络通信在当今高速发展的信息技术中占据着越来越重要的地位,作为其载体的世界上最大的通信网络——电信网的传输机制已从准同步数字序列(PDH)转向了同步数字序列(SDH)。SDH 网络的各项强大的功能和优越的性能要通过先进的管理系统或管理网络才能体现出来。本文首先论述了系统的需求要点,基于统一建模语言给出了用例图描述的分析结果,并对相应的场景进行了分析;接着利用对象建模方法及其它一些主流的面对象的技术对系统的建模进行了详细的分析与设计,包括从静态的类的结构分析和动态的类行为两方面进行了深入的研究,最后设计了系统的组件体系结构,本文的成果已在部分电信部门获得使用。

关键词 面向对象技术,电信管理网,同步数字序列,统一建模语言

Analysis and Design of SDH Network Management System Based on UML

SUN Zhi-Xin WANG Ru-Chuan WANG Shao-Di

(Department of Computer Science & Technology of Nanjing University of Posts and Telecommunications, Nanjing 210003)

Abstract Network Communication is playing more and more important role in modern information technology. As network communication's carrier, telecommunication network, the largest communication network, has finished transforming from PDH to SDH. Only by advanced management system or management network can the powerful functions and outstanding performance be manifested sufficiently. This paper firstly discusses the key points of the system's requirements, using the use case picture to express the results and then analyzing the relative scenarios. In succession, the Object-Oriented Methodology (OMT) method and other popular object-oriented methods are used to model the system in detail. The researches include both static structure analysis of classes and dynamic behavior of classes. Finally, a component system structure is designed for the system. The achievement of the system has been employed in several telecommunicate departments.

Keywords Object-oriented technology, Telecommunication management network, Synchronous digital hierarchy, Unified modeling language

1 引言

同步数字序列(Synchronous Digital Hierarchy, SDH)作为新一代的传输网体制,统一了世界上原来存在的两大互不兼容的数字体系,实现了数字传输机制上的国际标准。与准同步数字序列(PDH)相比,SDH 的一个显著特点是在帧结构中安排了丰富的比特开销,从而使其网络的监控和管理能力大大增强。SDH 的各项强大的功能和优越的性能正是在先进的管理系统之下才充分显示出来的。在 SDH 管理网中,为了提供一定的标准化方法来保证网管设计和定义的模块化特征,协议和过程的可扩展性,同时保证各厂家的产品能够实现纵向与横向的兼容性,采用了开放系统互连(OSI)的网络管理体系结构,在管理对象的实现中,应用了面向对象的分析与设计技术。

本文的工作基于信息产业部重点科研项目“TMN 各子系统互连方案的研究与实现”,对基于 TMN 的 SDH 网管系统进行分析设计,采用 UML 表示系统分析设计结果。UML 是一种标准建模语言^[1~3],而不是一种方法,因此现有的面向对象的方法都可以用 UML 的丰富图例来表示,同时 UML

支持多种面向对象的分析与设计方法^[6~8]。在系统分析过程中主要采用 OMT 方法^[4],同时借助于 OOSE 的用例需求分析方法^[5]及 UML 设计者所倡导的统一开发方法^[9]以及传统的数据流方法来说明面向对象技术在基于 TMN 的 SDH 网管系统开发中的具体应用。本文重点论述了与电话交换网网管、接入网网管等 TMN 网管子系统密切相关的故障管理和性能管理方面的分析和设计。首先进行了系统的需求分析,得出用例图,并对相应场景进行描述。然后进行系统的动态和静态分析,得出完整的类图,并以此为基础进行了系统的体系结构设计,最后给出了本文的结论。

2 系统描述与分析

传统的面向对象需求分析方法只给出了系统的对象模型,没有描述系统的功能需求,特别是从用户的角度来看待系统,为此,本文采用了广泛用于需求分析的用例(USE CASE)技术来描述用户所关心的系统功能,用序列图和协作图从不同侧面描述用例的各种场景的实现细节。

2.1 用例

用例^[1,2]就是“用户和系统在一次相互作用中相关事务的

孙知信 博士,研究方向:移动代理,组件技术等;王汝传 教授,博导,研究方向:分布式网络计算,移动代理等;王绍楦 教授,博导,研究方向:分布式体系结构,模式识别,TMN 等。

一个特殊序列”，用以描述用户与系统之间的相互作用，确定系统为用户提供的服务，从而得到准确的需求模型。用例图由一组被边界包围的用例、系统边界外的行动者或称操作者(Actor)以及操作者与用例间的通讯组成，是系统开发人员同用户进行交流的平台，主要刻画系统的功能需求和环境的约束，因此对于捕获用户的真正需求以及系统功能的扩充有着极为重要的用途。每个用例表示一种使用系统的方法，每个用例完成后生成不同的结果。本系统有四个顶层用例图，如图1所示，每个用例又有各自的用例子图，其中故障管理的用例如图2示。用例的描述如下。

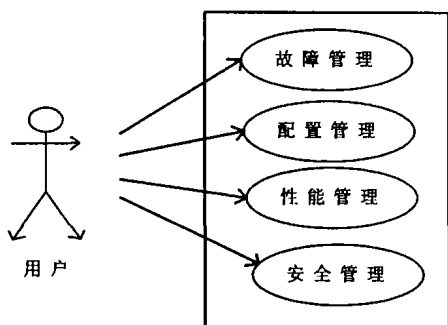


图1 顶层的用例图

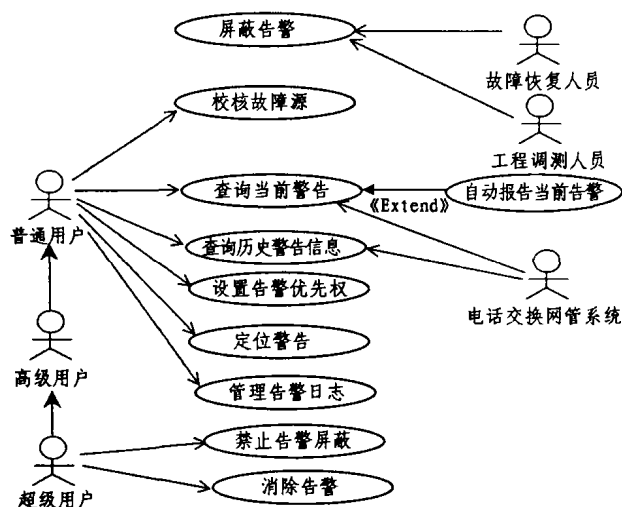


图2 故障管理用例图

•故障管理 用户使用告警监视，所谓告警就是作为某一事件、情况引起的结果由网元(Network Element, NE)自动产

生的指示。用户需要查看告警日志及告警的记录。用户通过网元管理系统的告警监视和界面显示，能把故障准确地定位在某一站某一子框的某一单盘上，或定位到某一 NE 的某个功能块上。同时用户测试接入、诊断和环回。测试的主要目的之一是故障诊断。向用户提供的测试工具应具有能按请求激活，以便检索从开销来的信息或操作特殊电路来支持故障分析的功能。

•性能管理 包括利用 SDH 结构有关的性能基元采集误码性能、缺陷和各监视项目数据；15分钟和24小时性能监视历史数据寄存和记录；门限设置和门限通知；性能数据分析和性能数据突破门限事件报告；不可用时间的起止记录和其间的性能监视等功能，如性能时间的处理、性能数据收集和日志处理等。

•配置管理 主要实施对 NE 的控制、识别和数据交换，实现对 NE 的指配调度功能以及 NE 状态和控制功能，其中一个主要的手段是实施保护倒换以控制、保护线路传输的业务不中断。

•安全管理 涉及注册、口令和安全等级等方面。防止未经许可而与 SNE 通信，保证安全地接入 SNE 的数据库。

2.2 场景描述

用例实例化后产生的实例——场景用来描述用例的动作序列的执行。可以用不同的场景刻画同一个用例实例。多个用例的多个场景建立系统的行为模型。为了刻画这些场景，UML 使用交互图对场景进行建模。

图3给出校核故障源的序列图。校核故障源是指从故障出现到正式入册之前，系统管理员即系统的用户应对故障源进行的校核操作。若故障持续时间在3秒以上就定位正式的传输故障，如果连续时间超过10秒该故障不再出现了，则认为故障已经消除或自动消失。在图中，用户首先向故障管理用户操作界面发出校核故障源请求(Check Fault Source)，用户界面给出被管 SDH 网络的所有当前的故障列表供用户选择；在用户选择了要校核的故障源之后，用户界面就向站点发送校核本站点故障源的消息(Check Station Fault)，然后站点向网元、网元向机盘逐级将校核故障源的消息传递到发生故障的机盘，由机盘来完成对故障源的校核，图中的 Check Fault 消息是一个反身消息它表示在规定的时间内对机盘进行反复校核。规定时间结束后，出故障的机盘将向用户界面返回校核的具体结果，最后由用户界面显示给用户，供用户分析该告警是否是真正告警，以及该故障是否已消除或自动恢复。

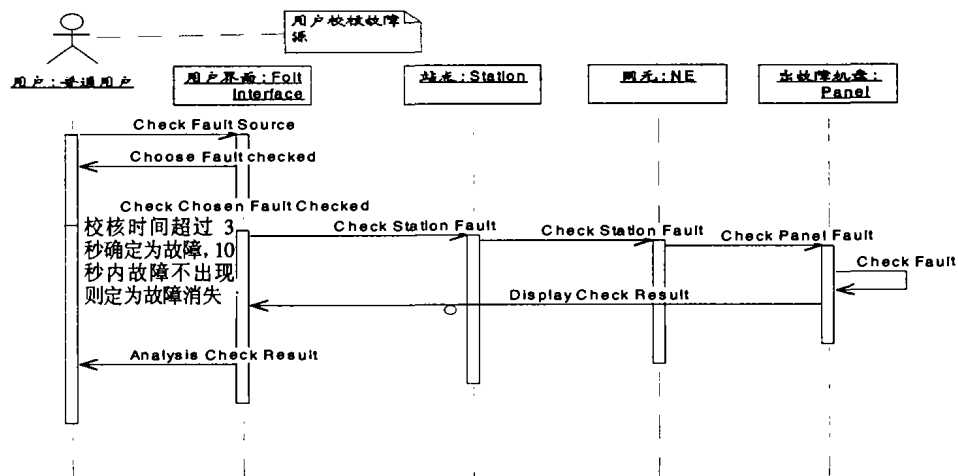


图3 故障源校核序列图

3 标识对象类

在需求分析之后,进行领域分析是关键的一环。根据已有的需求,给出问题的解决方案,以实现用例。经过对实际 SDH 网管系统进行的分析,依照 TMN 子系统互连的相关建议和标准,结合项目的具体需求,用 Power-Designor 绘制出了基于 TMN 的 SDH 网管系统的实体-关系图。考虑到所设计 SDH 网管系统在互连方面的重要需求,在设计这个实体-关系图的时候对与其它网管子系统包括电话交换网管和接入网管等的密切相关的实体:路径配置实体和链路配置实体进行了重点考虑,抽取添加了与电话交换网中的中继群的故障管理、性能管理以及配置管理等相关的属性以实现和电话交换网管系统的互连互通,使得接入网管和 SDH 网管能以电话交换网管为纽带,三个基于 TMN 的网管系统形成一个统一的基于 TMN 的综合网管系统。实体关系图可协助发现系统的潜在的对象类。在基于用例的分析和开发中,使用序列图及协作图的场景分析也可帮助发现对象类。同时分析每个对象及实体的责任可进一步缩小对象类的初始集合。经过以上几个方面的综合,得到了对象类集合及对每一个类的简短责任描述,部分对象类及其责任如表1所示。

表1 部分对象类及其责任描述

对象类名	对象类责任
站点	封装了系统向设备发出校核故障源等要求时必须进行的行政上所设立机构的授权等方面的服务。
网元	管理的核心,它封装了各种不同网元设备实现技术以及收集所包含的机盘上的告警向网管系统报告等方面的秘密。
机盘	封装了机盘所实现的设备功能、如何检测自身所发生的故障以及如何如何进行保护倒换等方面的细节。
交叉链接矩阵	封装了交叉连接矩阵所属网元上的各级通道如何与其它网元的通道进行交叉连接的信息。其中当然也包含了与交叉连接实体的指配相关的一些服务,因这些内容属于配置方面加入的服务这里不作详细讨论。

4 构造类图

4.1 标识子系统

首先,将整个网管系统分成故障管理、性能管理、配置管理和安全管理四个子系统,然后又将故障管理分成告警监视、故障定位、告警日志管理等子包,将性能管理分成性能事件管理、性能数据处理、门限管理和性能日志管理等子包,将配置管理分成了资源指配、资源状态控制和配置数据管理等子包。至于安全管理我们则将它作为一个子包处理。当然根据需要,我们还可以建立一些包,比如在告警监视中可以建立告警报告、告警分析等子包。包的功能划分与前面的用例模型之间有一定的联系,在具体实现时,一个包可能实现一个或若干用例的功能。

本系统将相关的对象类放在一个包中。经过分析得到的基于 TMN 的 SDH 网管系统子系统的包如图4所示,这里从软件设计的角度加上了用户接口包。经过这样的分“包”操作,我们就把完成故障管理的类和完成性能管理的类分隔开来,避免了因在一张图中显示太多的类而给分析人员带来管理和描述的不便,从而使得整个模型的层次结构更加清晰。

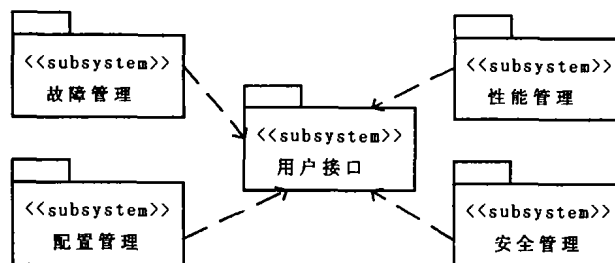


图4 SDH 网管系统的包图

4.2 定义类之间的关系

类之间的关系分为两种类型:一般-特殊(或称为泛化-特化)关系和整体-部分关系。前者建立了继承关系,而后者标识了组装关系。如果父对象的属性或特征可为其所有的子对象共享,就建立了泛化-特化关系结构。而如果父对象是由若干子对象以某种方式组装而成的,则就构成了整体-部分关系结构。整体-部分关系具有重复度和参与度特征。重复度是指组成一个父对象的子对象的数量。比如,一条路由由一个以上的链路(Link)组成。而参与度是指父对象或子对象是否都必须在整体-部分关系中出现。一条路径必须有一条链路来承载,而一个链路并不一定是路径的一部分,它可能还处于闲置状态。

从对象命名中常常可以发现一些整体-部分关系。如果一个对象与其它多个对象有大体相似的关系就有可能是一个泛化-特化结构。同时也可参考使用实体-关系模型来确认对象类之间的关系,比如,机盘和光通道机盘就是泛化-特化关系。经过以上过程得到的系统的类的关系结构如图5所示。

在标识结构时总会追加一些尚未被标识的类。就泛化-特化结构来说,它能很方便地标识出没有实例的父类。这样的类属类或称模板类只用于合并某些属性或服务,否则这些属性和服务就会在子类中被定义多次。比如性能(Prfm)就是这样一个模板类。

4.3 标识属性

在标识属性的过程中本文采用的基本做法是:标识属性、修改 OOA 模型中的其它特性以适应这些属性,如果有必要,可修改对象,然后重复这个过程。

由于作为预分析,已经建立了数据视图,即实体-关系图,其中实体的属性表示了存储数据的需求,这些存储数据必须出现在 OOA 模型中。从 ERD 图入手标识对象的属性,如实体可能对应着某一类。这样,实体属性就会简单地称为类属性。当然一个实体不一定只对应一个类,那么这个实体的属性必须分配到 OOA 模型的不同类之中,如机盘实体的属性就分配到了光通道机盘、微处理器机盘、电源机盘等几个对象当中。同样地,也可能将多个简单的实体的属性放在一个对象中。比如,链路状态、路径状态以及网元状态实体的属性就分别放到了链路、路径以及网元类中。

在标识对象属性的过程中,如果发现其某一属性的值是“不适用的”,就应重新考虑对象。所谓“不适用”就是指这个对象的所有属性不能完全应用于该类的所有实例中,即使其值为空。

4.4 标识实例关系

实例连接关系是 OOA 模型的属性中的另一重要内容,这些连接指明一个类中的对象是如何与另一个类中的对象相连的。当这些对象类被实现的时候,这些事务规则就控制服务

工作,并确保与所建立系统的目的相一致。标识实例关系即在前面建立的模型的基础上,为基于 TMN 的 SDH 网管系统的类的实例建立一系列的实例连接,为每个实例连接定义重复度(1:1,1:M,M:M,即一对一、多对一还是多对多)和参与度

(可选的还是必需的)。

最后我们得到配置管理及性能管理的类图,图5为系统部分类图。图中显示了类之间的关系、实例连接及属性与方法等。类图中的类分属于两个包。

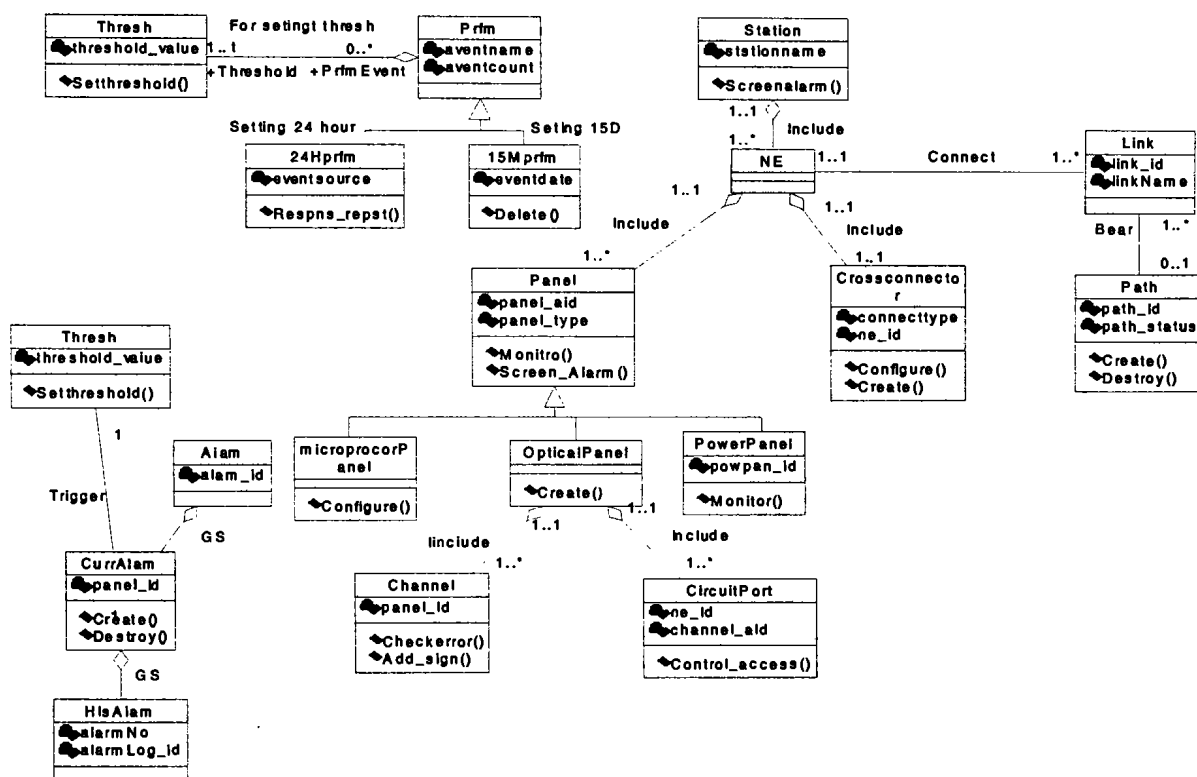


图5 基于 UML 的 SDH 网管系统的类连接关系

5 定义对象动态模型

到现在为止 OOA 模型中的每一成分表示的都是应用系统中的某个静态方面。比如说,把网元纳入与机盘的整体-部分关系,但忽略了网元如何才会出现故障状态以及性能门限超越事件发生后如何报告相应的通道和机盘的故障等方面的细节,为此需要建立对象之间的动态关系。它们可以表示为对象所执行的服务以及对象之间传递的消息。

给模型中的每个对象分配合适的服务好像很简单,其实由于系统的 OOA 工作已进行到相当深的层次,各个对象及其关系已相当复杂,因此在决定某些服务的最终归属问题时仍需要进行全盘细致的考虑,在多种可能的方案中作出最佳选择。在建立动态模型的过程中,我们首先标识了在每个类中必须封装的一组服务;然后将服务与类的属性相比较,验证其一致性。对于标识出的属性中没有关联到任何服务的,对其进行删除,因为这些属性对于类来说形同虚设,不可能被访问。然后,我们画出了类之间的消息路径以研究系统的行为。这一过程的一种做法是:访问每一个类,向每个类提出这样的问题“你是如何建立的?你的责任是什么?你下一步要做什么?……”,从而给出了许多从对象建立到消亡的状态。每一状态的改变都关联到对象之间消息的接受(及传递)。这是一种自底向上的方法,从对象着手,逐渐向上分析。另外一种方法是采用事件-响应方法。对象必须识别出系统中可能发生的所有事件,然后产生发送给其它对象的消息,最后由那些对象产生响应。这样就能够找出对象所必须执行的服务以便接收、处理以及产生每个消息。这是一个自顶向下的方法。本文采用以前者为主,后者为辅的方法,因为前者简单明了,实施起来较为

直观,而后者是从系统的行为入手,然后逐渐分析到对象,有利于更好地满足用户需求。利用 UML 我们建立了一些复杂、关键类的状态转移图。

6 建立系统的体系结构模型

在我们识别了所有对象之后接下来的一个问题就是如何将对象分配到相应的工程文件或动态链接库中。在 UML 中提供了一种组件图机制,用以实现对象类的分配工作。我们用一个组件表示一个软件模块,同时也表明了前面我们分析所得到的这些类和对象的具体实现方式,具体做法如下:

- 基于上述的分析与设计结果,首先划分出用于实现各个类和对象的软件模块,定义出组件;软件模块之间的关系就是组件之间的调用和依赖关系。将各种应用共同使用的资源对象按功能组织成铅版(Stereotype)为 DLL 的组件,而将与特定应用有关的对象组织成铅版为 Application 的组件。

- 系统软、硬件的物理配置应依据所采用的软件环境来定。例如,如果采用 B/S(Browser/Server),那么用于实现类和对象的软件模块(或组件)都放在服务器端,客户端则不放置任何组件。

在数据库模型的设计方面,有几种设计方法可选。一种是在分析与设计的基础上,利用 Rational ROSE 提供的与 ERwin 的接口,在类的定义窗口中的 ERwin 的接口定义域内对相关项进行准确的定义,然后生成相应的后缀名为 .rdx 的 ERwin 文件,借助 ERwin 就可以生成 E-R 图,从而进一步生成数据库的物理模型,并在服务器建立相应的数据库和表。第

(下转第58页)

- 19]. UMBC, December, 2001. <http://research.ebiquity.org/re/papers.html>
- 7 Chakraborty D. Service Composition in Ad-hoc Environments: [Technical Report TR-CS-01-20]. UMBC, October, 2001. <http://research.ebiquity.org/re/papers.html>
- 8 Gunther C W. Implementing Interorganizations Workflows via Web Services Orchestration. Hasso Plattner Institute for Software Systems Engineering, Germany. 2002
- 9 DAML Joint Committee. Daml + oil (march 2001) language. <http://www.daml.org/2001/03/daml+oil-index.html> 2001
- 10 DAML Joint Committee. Daml Services (Daml-S) May 2003. <http://www.daml.org/services/> 2003
- 11 Fensel D, Bussler G. The Web Service Modeling Framework WSMF. Electronic Commerce Research and Application. 2002. 113~137
- 12 Miller J A, et al. Composition, Performance Analysis and Simulation of Web Services. <http://lsdis.cs.uga.edu/lib/download/CMSA+02.pdf> 2002
- 13 Bierhoff K. Web Service Composition Patterns. Web services for B2B Integration-report on the seminar. Hasso Plattner Institute for Software Systems Engineering, Germany. 2002
- 14 Leymann, F. Web Service flow language (WSFL) 1.0. <http://www-4.ibm.com/software/solutions/webservices/pdf/WSFL.pdf> 2001
- 15 Paolucci M, Kawamura T, Payne T R, Sycara K. Importing the Semantic Web in UDDI. <http://citeseer.nj.nec.com/540826.html> 2002
- 16 Miller J, Nair R, Zhang Z, Zhao H. JSIM: A Java-based Simulation and Animation Environment. In: Proc. of the 30th Annual Simulation Symposium, Atlanta, GA, 1997. 31~42
- 17 Narayanan S, McIlraith S. Simulation, Verification and Automated Composition of Web Services. In: Proc. of the Eleventh Intl. World Wide Web Conf. Honolulu, HI, 2002. 77~88
- 18 Dussel P. Composition of e-Services in a highly dynamic environment. Hasso Plattner Insititue, Germany. 2002
- 19 Hamadi R, Benatallah B. A Petri Net-Based Model for Web Service Composition. In: 14th Australasian Database Conf. (ADC2003), Adelaide, Australia. Conferences in Research and Practice in Information Technology, Vol. 17, 2003
- 20 Semantic Web Service Architecture—Evolving Web Service Standards toward the Semantic Web. American Association for Artificial Intelligence, 2001
- 21 McIlraith S A, Son T C, Zeng H. Semantic Web Services. IEEE Intelligent System, 2001
- 22 Tidwell D. Web Services: The Web's Next Revolution. <http://www-106.ibm.com/developerworks/webservices/> 2000
- 23 Thatte S. XLANG: Web Services for Business Process Design. <http://www.gotdotnet.com/team/xml-wsspecs/xlang-c/default.htm> 2001
- 24 UDDI Spec Technical Committee Specification. UDDI Version 3.0, July 2002. <http://www.uddi.org/pubs/uddi-v3.00-published-20020719.htm> 2002
- 25 W3C. Soap version 1.2 Proposed Recommendation, May 2003. <http://www.w3.org/TR/soap12-part0/> <http://www.w3.org/TR/soap12-part1/> <http://www.w3.org/TR/soap12-part2/> 2003
- 26 Web Services Description Working Group. Web Services Description Language (WSDL) Version 1.2, March 2003. <http://www.w3.org/TR/wsdl12/> 2003
- 27 Waldt D, Drummond R. EBXML: The Global Standard for Electronic Business. <http://www.ebxml.org/presentations/global-standard.htm> 2001
- 28 Zeng L, Benatallah, B, Nguyen P. AgFlow: Agent-based Cross-Enterprise Workflow Management System. In: Proc. 27th Intl. Conf. on Very Large Data Bases, Italy, 2001
- 29 Gray J, Paper I. The Transaction Concept: Virtues and Limitations. In: Proc. of the 7th Intl. conf. on Very Large Data Bases, Sep. 1981. 144~154
- 30 Lynch N A, Merritt M, Weihl W E, et al. Atomic Transactions. Morgan Kaufmann, ISBN 1-55860-104-X, 1993
- 31 Bernstein P A, Newcomer E. Principles of Transaction Processing for Systems Professionals. Morgan Kaufmann, ISBN 1-55860-415-4, 1996
- 32 Paulo de Figueiredo Pires. WEBTRANSACT: A framework for specifying and coordinating reliable web services compositions. <http://www.cos.ufrj.br/~pires/webTransact.pdf/pires02webtransact.pdf> 2002
- 33 Graham S. Building Web Services with Java-- Making Sense of XML, SOAP, WSDL, and UDDI. China Machine Press, 2003
- 34 BizTalk Server Homepage. <http://www.microsoft.com/china/biztalk/default.asp>

(上接第51页)

二种设计方法就是利用 Rational ROSE 提供的数据库描述语言(DDL)生成功能,首先在类的定义窗口中的 DDL 接口定义域内定制库表的列名、列的类型以及主键等方面的细节,然后利用 ROSE 来生成包含可执行 SQL 语句的后缀名为.sql 的文件,借助于它就可以建立相应的数据库表。第三种方法是通过在前面所述的 E-R 图上作一定的修改来完成数据库的设计。鉴于目前本项目开发时没有 Erwin 软件包,同时采用 ROSE 生成 DDL 的方法较难生成相应的实体-关系图、物理模型图和相关文档,而 Power-Designor 则在逻辑模型的建立、逻辑模型到物理模型的转换以及各种设计文档的生成等方面则具有强大的功能,所以采用了第三种方法进行了数据库的具体设计。

结论 本文从面向对象的角度全面论述了基于 TMN 的 SDH 网管系统的分析与设计。UML 是一种通用的建模语言,因此本文的分析结果都用其进行描述,但 UML 不是一种方法,因此面向对象中已有的方法将继续得到应用。在系统的开发过程中,我们注意到不能盲目地抛弃一些实践证明有效的一些传统方法在面向对象开发中的应用,如数据流方法及实体-关系图,本文的成果是一种成功的尝试,对电信网管系统

的面向对象的应用具有积极的参考价值。

参 考 文 献

- 1 Booch G, Jacobson I, Rumbaugh J. UML Notation Guide (Version 1.1). Rational Corporation, Santa Clara, 1997
- 2 Fowler M, Scott K. UML Distilled: Applying the Stand Object Modeling Language. Reading, MA: Addison-Wesley, 1997
- 3 Object Management Group. OMG Unified Modeling Language Specification, v. 1.3 June 1999
- 4 Rumbaugh, James, et al. Object-Oriented Modeling and Design. Upper Saddle River, NJ: Prentice Hall, 1991
- 5 Jacobson, Ivar, et al. Object-Oriented Software Engineering: A Use Case Driven Approach. Reading, MA: Addison-Wesley, 1992
- 6 Fowler M, Scott K. Applying the Stand Object Modeling Language. Addison-Wesley, Reading, MA, 1997
- 7 Filippidou D. Designing with scenarios: a critical review of current research and practice. Requirements Engineering, 1997, 3: 1~22
- 8 Richter C. Designing Flexible Object-Oriented Systems with UML. Macmillan Technical Publishing, Indianapolis, IN, 1999
- 9 Jacobson I, Booch G, Rumbaugh J. The Unified Software Development Process. Addison-Wesley, Reading, MA, 1999