应用技术

水资源供需平衡评价系统软件 UML建模

黄 书 汉
(西北大学 地质系, 陕西 西安 710069)

摘 要: 针对实例引入基于面向对象的 UML标准建模语言进行水资源供需平衡评价系统分析和系统设计;用 Rational Rose进行系统 UML软件模型的建模;抽象出一些实例对象的 C++类,用 Visual C++开发了32位中文版 Windows平台上的水资源供需平衡评价软件等一系列工作。

关键词: UML; 水资源; 供需平衡; 面向对象; 软件; Visual C++ 5.0; Windows
文献标识码: B 文章编号: 1000-288X(2001)06-0048-05 中图分类号: TV213, TP39

UML Modeling Software of Evaluating System of Balance Between Supply and Demand in Water Resource

HUANG Shu-han
(*Geology Department of Northwest University, Xi'an 710069, Shaanxi Province, PRC*)

Abstract The UML was introduced to modeling the real software for evaluating the balance between supply and demand in water resource system in its analysis and design by using Rational Rose, drawing out some C++ Class of instance from the real system. Its software was finally come true by using Visual C++ 5.0 on Win32 technique in Windows95.

Keywords UML; water resource; the balance of supply and demand; object-oriented; software; Visual C++ 5.0; Windows

面向对象理论直接模仿人们解决现实世界实际问题的思路和方法,可用于编制具有高度可重用、可扩展、可维护和易操作性的软件。水资源系统属于复杂系统,涉及变量众多,变量间关系错综复杂,工作量大,其软件开发必须引入面向对象的系统工程思想,首先进行系统分析和设计,不能直接进行程序模块编写,否则达不到预期目的。统一建模语言(UML: Unified Moding Language)[1,2],或称标准建模语言,是面向对象技术发展过程中的重要成果,是适合各种系统建立模型的通用语言。作者尝试应用软件工程技术和UML技术,用 Rational Rose进行水资源供需平衡评价软件的系统建模,用 Visual C++进行开发。

1 UML简介

1.1 标准建模语言 UML的主要特点

UML是一个通用的标准建模语言,其目标是以面向对象图的方式来描述任何类型的系统;可以对任何具有静态结构和动态行为的系统进行建模;具有很宽的应用领域。其中最常用的是建立软件系统的模型。但它同样可以用于描述非软件领域的系统。

在建立软件系统的模型时,UML适用于系统开发的过程中的不同阶段。

(1) 业务建模阶段: 通过用例建模,描述业务系统,需要识别问题域中的重要概念和机制。

(2) 需求分析阶段: 通过对象(类图)建模,描述对系统感兴趣的外部角色及其对系统运行的理解。

(3) 分析阶段: 用 UML类图或包含类的具有层次结构的包图描述系统对象及其相互间的关系,建立系统的软件模型。

(4) 设计阶段: 定义软件系统中类(如处理用户接口、数据库、通讯和并行性等问题的类)的更详细的规格说明,并且,对于可分布的系统,说明类间关系及其所在的可分布的组件。

(5) 编程(即构造或实现)阶段,是一个独立的阶段,用面向对象编程语言将来自设计阶段的类转换成实际的代码,并集成为组件。

(6) 测试阶段: UML模型可作为工作依据,不同的测试小组使用不同的 UML图。单元测试使用类图和类规格说明; 集成测试使用配置图、组件图和合作图;系统测试使用用例图来验证系统的行为;验收测

收稿日期: 2001-08-28
资助项目: 国家"九五"攻关项目"西北地区水资源合理利用与生态环境保护研究"(96-912, 96-912-05-02-02-01)
作者简介: 黄书汉(1964-),男(汉族),陕西省宝鸡县人,现在东北大学做博士后研究。主攻流体资源系统分析与可视化模拟技术。E-mail: hshl@sina.com

试由用户进行,以验证系统测试结果是否满足分析阶段确定的需求。

在用 UML建立分析和设计模型时应避免考虑把模型转换成某种特定的编程语言。因为在早期阶段,模型仅仅是理解和分析系统结构的工具,过早考虑编码问题十分不利于建立简单正确的模型。

1.2 标准建模语言 UML的主要内容

UML由 5类图 (共 9种图形)来定义。

(1) 用例图 (Use- case diagram): 从用户角度描述系统功能,并指出各功能的操纵者。

(2) 静态图 (Static diagram): 包括类图、对象图和包图。类图描述系统中类的静态结构,定义系统中的类,表示类之间的联系 (如关联、依赖、聚合等)并包括类的内部结构 (类的属性和操作),在系统的整个生命周期都是有效的;对象图是类图的实例,存在生命周期,因此只能在系统某一时间段存在;包图由包或类组成,表示包与包之间的关系。包图用于描述系统的分层结构。

(3) 行为图 (Behavior diagram): 描述系统的瞬态动态模型,包括状态图、活动图等。状态图描述系统的对象所有可能的状态以及事件发生时状态的转移条件;活动图描述满足用例要求所要进行的活动以及活动间的约束关系,有利于识别并发活动。

(4) 交互图 (Interactive diagram): 描述对象间的交互关系,包括顺序图、合作图等。顺序图显示对象之间的动态合作关系、强调对象之间消息发送的顺序及其交互作用;合作图描述对象间的层次协作关系。如果强调时间和顺序则使用顺序图,如果强调上下级关系则选择合作图。这 2种图合称为交互图。

(5) 实现图 (Implementation diagram): 包括构件图、配置图等。构件图描述代码部件的物理结构及各部件之间的依赖关系,包含逻辑类或实现类的有关信息,有助于分析和理解部件之间的相互影响程度;配置图定义系统中软硬件的物理体系结构。

从应用角度看,当采用面向对象技术设计系统时,首先是根据业务模型描述出系统需求;其次根据需求建立系统的静态模型以构造系统的结构;第三步是描述系统的行为。其中在第一步与第二步中所建立的模型都是静态的,包括用例图、类图 (包括包)、对象图、部件图和配置图 5种图形,是标准建模语言 UML的静态建模机制。其中第三步中所建立的模型或者可以执行,或者表示执行时的时序状态或交互关系。它包括状态图、活动图、顺序图和合作图等 4种图形,是标准建模语言 UML的动态建模机制。

总之,标准建模语言 UML适用于以面向对象技术来描述的任何类型的系统,而且适用于系统开发的不同阶段。

1.3 业务建模与业务用例模型

在面向对象开发和传统的软件开发中,人们根据典型的使用情景来了解需求。用例模型是第二代面向对象技术的标志,主要描述的是外部执行者 (Actor)所理解的系统功能。

业务建模阶段中建立的业务用例模型是系统开发者和用户反复讨论的结果,表明开发者和用户对业务领域达成的共识。

获取用例首先要找出系统的执行者。可通过用户回答: 谁使用、维护、管理系统或系统的部分功能?谁需要系统来工作?系统需要哪些硬件?需要与哪些系统或应用程序交互?等问题来识别执行者。执行者未必是人,也可以是一个外界系统,可能需要从当前系统中获取信息,或与当前系统进行信息交互。

对获取的每个执行者或整个系统提出问题就可以获取用例。如要求系统提供那些功能?产生、删除、修改或存储的信息有哪些类型?必须提醒执行者的系统事件有哪些?执行者的某些典型功能能否被系统自动实现?系统需要何种输入输出?输入从何而来?输出到何处?当前系统的主要问题?等等。对于大系统,先列出执行者清单再分析,获取用例就会很容易。

几乎在任何情况下都会使用用例。用例用来获取需求,规划和控制项目。用例的获取是业务建模和需求分析阶段的主要且首先要做的工作。大部分用例在项目的业务建模阶段产生,并随工作的深入细化,应及时增添。业务用例集里的每个用例都是一个潜在的需求。一般情况下,业务对象模型隐含在业务用例的顺序图中,对于大系统或流程多变的系统,有必要单独形成特定的模型。

1.4 需求分析与对象模型

需求分析阶段分析描述待开发系统的体系结构,形成考虑了现有资源条件和各方利益的系统用例分析模型;从用户的角度来理解系统;驱动以后各阶段的开发工作,影响 UML的各个模型。

1.5 类图技术介绍

类 (Class)是对一类具有相同特征的对象的描述,而对象是类的实例 (Instance)。类 (Class)、对象 (Object)和它们之间的关联是面向对象技术中最基本的元素。类模型和对象模型揭示系统的结构。在 UML中,类和对象模型分别由类图和对象图表示。类图技术是 OO方法的核心。建立类模型时,我们应尽量与应用领域的概念保持一致,以使模型更符合客观事实,易修改、易理解并易交流。类图 (Class dia-

gram)描述类与类之间的静态关系。与数据模型不同,它不仅显示了信息的结构,同时还描述了系统的行为。类描述一类对象的属性(Attribute)和行为(Behavior)。在 UML中,类的可视化表示为一个划分成 3个格子的长方形。最顶的格子中是类的名称,中间的是属性,最下的是操作项。

类的名称一般是名词,其获取必须与研究领域内的专家合作,尽量采用术语,应明确、无歧义。

用类的属性来描述并包括能区分该类每个特定的对象的,或是系统感兴趣或建模目的所要求的共同特点。属性项可以包括可见性、属性名称、类型、缺省值和约束特性,也可省略每项。UML规定属性的语法为:可见性 属性名:类型 = 缺省值 {约束特性}。属性的可见性即是否对外界隐蔽。类型表示该属性的种类,可以是基本数据类型,也可以是用户自定义的类型。约束特性则是用户对该属性可操作性限定的说明。例如"{只读}"说明该属性是只读的。

类操作即修改、检索类的属性或执行某些动作,可省略。UML规定其语法为:可见性 操作名(参数表):返回类型 {约束特性}。操作名、返回类型和参数表组成类的操作界面。

类图描述类和类之间的静态关系。定义了类之后,就可以定义类之间的各种关系了。

关联表示两个类之间存在某种语义上的联系,如"属于"、"签定"。人们将具有共同特性的元素抽象成类别,并通过增加其内涵而进一步分类。继承定义了一般元素与特殊元素之间的分类关系。有 2个元素 X, Y,如果修改 X的定义可能会引起对另一个元素 Y的定义的修改,则称元素 Y依赖(Dependency)于元素 X。在类中,依赖由各种原因引起。

在软件开发的不同阶段都使用类图,表示不同层次的抽象。在需求分析阶段,类图是研究领域的概念,其模型应独立于实现它的软件和程序设计语言。

在设计阶段,类图描述类与类之间即软件的接口。可以用一个类型(Type)描述一个接口,这个接口可能因为实现环境、运行特性或者用户的不同而具有多种实现。在很多时候,此层的类图更易于开发者之间的相互理解和交流。

在实现阶段,类图描述软件系统中类的实现,才真正有类的概念,并且才揭示软件的实现部分。这可能是大多数人最常用的类图。

理解 3个层次对于画类图和读懂类图都是至关重要的。画类图时,要从一个清晰的概念层次观念出发;而读图时则要弄清它是根据哪种层次观念来绘制的,这对建模或者评价模型非常有用。

2 水资源供需平衡评价的主要内容

根据"江河流域规划编制规范(SL201– 97)"第 5条,水资源供需平衡分析与预测工作主要包括:按水资源供需系统分区,研究水资源数量和质量及其特点,分析其利用现状与存在问题,预测今后一定时期内各区的国民经济指标与相应的需水、耗水情况,估算分区水余缺量,并结合各区耕地、人口、工农业发展和水资源利用率等指标评价缺水地区的缺水性质、程度,提出合理利用水资源的工程措施和应采取的方针政策等项建议。

由水资源供需平衡评价的过程看,可分为以下几项主要内容:水量、水质的评价;水工程及规划的调查(供水条件);不同水平年或保证率的可供水量计算;需水量预测(工农业及生活用水总量);供需平衡计算;解决供需矛盾的对策研究。其实际运行流程为:选择计算年;水资源分区;准备(可供水量与需水量预测):供需平衡计算;对策研究。其中供需平衡计算为主要内容。

3 实例概况

水资源供需平衡系统一般涉及许多水工程,其相应的状态参数也有不同的界定理由。实际水资源系统的水工程组成不同,相应的供需平衡分析目的、方案不同,其评价软件的对象分析和系统设计也随之而不同。必须对其进行综合分析后,确定 UML模型。

3.1 实际条件

宝鸡峡灌区系统是一个由多种水利设施构成、联合运用关系复杂的系统,受到河道含沙量、干渠引水能力、灌区用水和水库汛限水位等因素的制约。此次水资源供需平衡评价的目的在于摸清供水满足需水的过程及程度、为水资源利用规划提供依据、为下游断面提供设计来水量数据。水资源供需平衡计算包括来水、用水和供需平衡关系分析三大部分内容。工作导则规定:按逐日平均流量和需水量进行供需分析;对引水沙限:引水工程按 15%,蓄水按 3%;利用现行灌溉制度及水利用系数确定需水量。按加闸与否,来水为实测的或还原扣水后得来的或为平均流量,满库或空库起调等多方案进行分析计算。

3.2 实例系统运行原则

以各种方案比较和最大限度节水用水为前提,确定系统运行原则如下。

(1)用水和蓄水原则:在灌溉季节,先满足灌区用水,再向水库充水;非灌溉季节,以相关设施能力来控制最大限度地拦蓄和引蓄河道弃水;若有渠首库,

应按“先尾库 (灌区库)后首库 (加闸库)”的次序 ,保证蓄水的安全性和有效性。

(2) 供水原则: 先引河道天然来水 ,后取水库蓄水;灌区和渠首均有库时 ,“先首库后尾库”。

(3) 沙限控制原则: 干渠引水和首库蓄水沙限为 15% ,灌区库蓄水沙限为 3%。均由统计资料转换为流量限制。

(4) 汛期运行原则: 已蓄库水位大于汛限水位时 ,首库弃水加进河道弃水 ,合并为下游来水量;尾库弃水因实际系统暂无出路 ,只作统计 ,不予分析。

(5) 逻辑结构确定原则: 以逐日时序为引导 ,灌区供水满足为控制 ,最大限度利用水资源为目的。

4　实例系统的 UML模型

实例系统的 UML模型系采用 Rational Rose进行系统的 UML软件模型建模工作 ,文中插图主要系根据拷屏制作图修改而成。

4.1　用例图

由于此次工作重点在于平衡评价部分 ,来水和需水部分的数据由其他工作人员计算完成 ,以纯文本文件为接口。考虑此次实际系统的水工程现状以及研究要求 ,经过面向对象抽象及分析 ,归结出根据实际系统的水工程现状以及研究要求 ,确定系统的用例图 (详见图 1)。

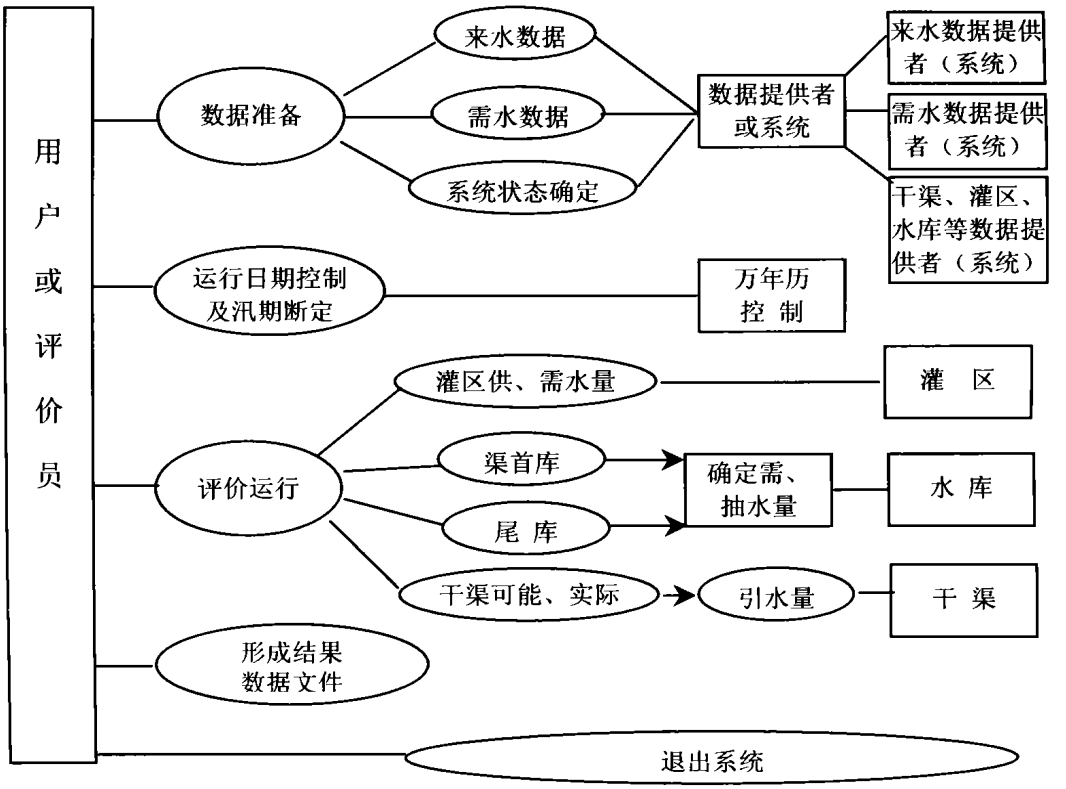

图 1　用图例

4.2　系统的静态图

此次水资源供需平衡评价的实际系统由河流、渠首规划库、引水干渠、灌区和灌区库等工程组成。

一般地 ,水库工程的特性指标有: 控制流域面积、调节性能、总库容、防洪库容、兴利库容、调节水量、调节系数、坝型;配套工程指标有: 灌溉面积、发电机装机容量、年发电量等。这里 ,根据实际研究需要 ,选定水库特性指标。

水库的功能或运行规则依据规范规定和实际需求分析来确定。类似地 ,对系统的其它工程如渠道等 ,进行面向对象分析。考虑此次实际系统的水工程现状以及研究要求 ,经过对象原型化抽象及分析 ,归结出 3种对象类: 水库类、渠道类、灌区类。通过系统对象分析抽象出系统的主体结构图称为包图。

4.3　系统的行为图与合作图

通过系统对象分析抽象出系统的行为图。限于篇幅 ,只给出灌区状态图和水库活动图 (图 2)。

系统的合作图描述对象间的交互关系: 顺序关系和协作关系。实际系统的合作图 (图 3)。这里 ,干渠为系统的中心和纽带;渠首库是拟建工程;箭头编号表示调用顺序 (1— 5属于准备阶段; 6— 8属于运行阶段),线上所注为协作量。

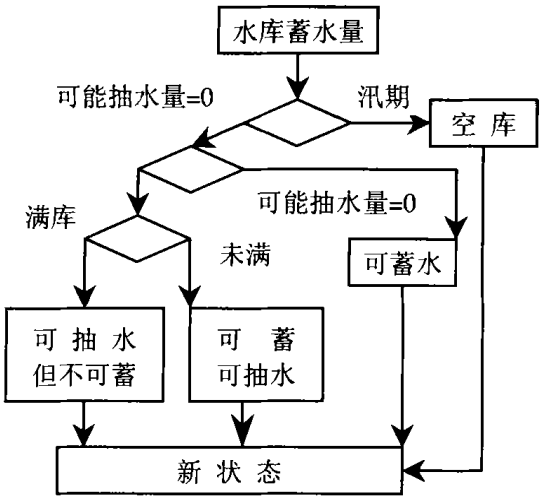

图 2 水库活动图

4.4 系统的输入输出设计

系统输入部分以“数据读入”模块形式处理;系统输出部分也同样以“形成结果文件”的数据存入模块形式处理。方案选取、各对象的状态参数输入、来水和需水数据文件名输入都在主界面上进行;修改了默认值后,必须按“常数设定”按钮进行确认。

按照水资源供需平衡评价系统设计的要求,除为了检查而形成的包括注释消息的内部临时文件以外,形成的最终结果文件由 3个文本文件组成: 即弃水数据文件,为纯数字数据文件,无冗余信息;以年为单位和以月为单位运行结果统计数据文件,为表格文件,并带有根据选定方案自动形成的表头和各个表项。表 1为采用了“加闸、满库起调”方案情况下的、以月为单位运行的结果统计数据文件的部分表格 (仅据示意性质)格式。

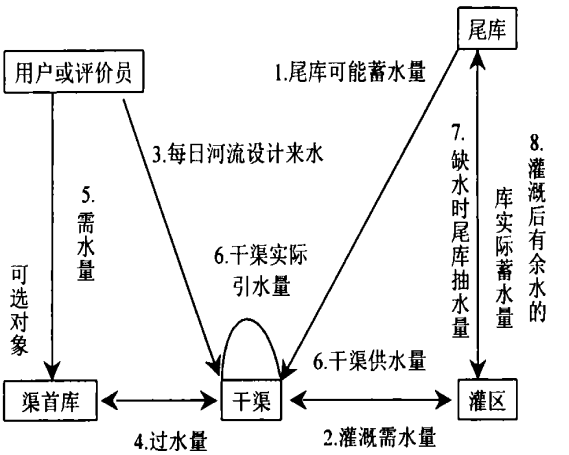

图 3 对象协作图

表 1 宝鸡峡塬上灌区水资源供需分析运行结果 10^4 m^3

月份	设计来水	灌溉需水	干渠引水	灌区用水	渠首库抽水量	渠首库蓄水量	灌区库抽水量	灌区库蓄水量	灌区缺水量	渠首弃水
1	4 859. 1	0. 0	0. 0	0. 0	0. 0	0. 0	0. 0	0. 0	0. 0	4 859. 1
…	…	…	…	…	…	…	…	…	…	…
12	6 795. 4	12 130. 6	9 572. 09	12 130. 6	2 844. 9	67. 4	2 772. 8	215. 1	0. 0	0. 0
合计	145 166. 1	71 539. 2	60 377. 5	62 935. 1	10 749. 1	13 949. 1	16 820. 2	14 262. 5	8 604. 0	87 988. 6

注: ① 1974年数据运行结果,表示渠首加闸情况;② 该灌区库各月防汛弃水皆为 0

5 基于 UML模型的软件开发

在运行时,用一个 ActiveX控件 (万年历)来实现逐日控制运行;用内部制约机制实现事件的正确执行顺序;为使用户易于学习,特地利用 Lotus公司 ScreenCam软件制作了“使用方法演示”的电影。

经过各种方案的运行验证证明: 我们建立的水资源供需平衡评价 UML软件模型及由此开发的软件系统性能稳定,易于使用,达到了预期的目标。

此次研究实践,将基于面向对象技术的标准建模语言 UML成功地引入水资源供需平衡评价中进行系统建模,由其开发的软件系统控制准确,运行良好。此次工作证实了基于面向对象技术的标准建模语言 UML是复杂系统问题建模的最好方法和工具,是实现实际系统数值模拟必备的基础工具。

[参 考 文 献]

[1] 中国计算世界机报编辑部. 技术专题,标准建模语言 UML及其支持环境 [N]. 1998.

[2] 李留英,等. 统一建模语言 UML[J]. 计算机科学,1998, 25(5): 20- 26