

软件开发

软件过程模型

管理信息系统

(23)

MIS

计算机科学1997Vol. 21No. 6

MIS 软件开发的过成模型

The Process Model of MIS Software Development

97-99

顾明

TP311-S2

TP399

(深圳高等职业技术学院计算机系 深圳 518055)

仲萃豪

(中国科学院软件研究所 北京100030)

摘要 A description of software process model using activity concept is presented. Based on this description, the process of MIS software development is analysed. A process model about MIS software development is given.

关键词 Software process; Model; MIS; Software development

自从 L. Osterweil 在1987年提出软件过程也是软件的思想之后,对软件过程的研究已引起了软件工程研究者的极大兴趣,开展了许多这方面的研究,归纳起来,主要有以下几个方面:建立各种形式的软件过程模型;研究软件过程的描述语言;开发支持软件过程模型和描述语言的软件工程环境。

本文是从 MIS 软件开发的需要来研究软件过程模型,我们提出的软件过程模型,力图克服 MIS 中常用的瀑布模型的缺点,描述角度是从行为而不是从算法上来描述软件过程,即用活动而不是用步骤来描述过程,因为过程本身就是一种活动,活动可以并行而步骤只能串行,因此活动比步骤更贴切地反映软件过程的本质。从 MIS 软件开发的角度看,为了正确地获得用户需求,人的认知过程有一个反复的过程,活动可以方便地表达这种反复性。

1 软件过程描述的符号说明

SPM (Software Process Model), 软件过程模型

Activity: 活动

Action: 动作

Precondition: 前置条件

Postcondition: 后置条件

Message: 消息

Composite-activity: 顺序组合活动

Complex-activity: 并行组合活动

Message-body: 消息体

Message-event: 消息事件

Sender: 消息发送者

Receiver: 消息接收者

{ } : 集合符号

() : 元组符号,其中每个元素以“,”分隔

::= : 定义符号,表示左边被定义为右边

< > : 定义中的每个成分,可以出现在定义符号的两边

△ : 并行符号,表示左右的二个活动并行执行

| : 选择符号,表示或者选择左边或者选择右边

Entity: 实体

2 软件过程的形式化描述

软件过程模型由一个活动集合来描述:

$$SPM = \{Activity\} \quad (1)$$

一个活动可以由更小的动作组成:

$$Activity = (\{Precondition\}, \{Action\}, \{Postcondition\}, \{Message\}, \{Entity\}) \quad (2)$$

说明:每个活动由一个5元组组成,其中包括:前置条件集合、动作集合、后置条件集合、消息集合和实体集合。活动可以在不同的地点由不同的人员来完成,可以串行,也可能并行。

前后置条件虽然类似于 Hoare 公理系统中的谓词,但对它们的解释有所不同。在我们的描述中,前后置条件表示软件过程的各种属性和软件活动的各种联系,而这种属性和联系可以是文档和资源等,也

可以是活动的输入或输出信息。一个动作可以有多个前置条件,在一个动作发生时,所有涉及该动作的前置条件都应满足;一个动作也可以有多个后置条件,当该动作完成时,可以只有一个后置条件被满足,到底哪一个后置条件被满足将取决于该动作所产生的结果。例如,涉及到一致性检查的一个动作,有二个后置条件,正确或错误,当检查的结论是成功时,后置条件“正确”被满足,当检查的结论是失败时,后置条件“错误”被满足。

消息提供在活动之间通讯的一种手段,也可以是与项目有关的里程碑,例如:一个消息标志一个活动已经完成,这是一个简单消息,也可以是复杂消息,如一个从一个活动到另外一个活动通讯的软件对象。我们如下定义消息:

$$\text{Message} = (\text{Message}, \text{Message-event}) \quad (3)$$

说明:一条消息是一个二元组,由消息对象本身和消息事件二部分组成,而消息事件又可定义如下:

$$\text{Message-event} = (\text{Sender}, \{\text{Receiver}\}) \quad (4)$$

说明:消息事件是一个二元组,其中包括消息发送者,消息接受者集合,这说明一条消息可以发送给多个接受者,即一个发送者可给多个接受者发消息。

实体是活动作用的对象,在不同的活动中,实体是不同的,在本文中,实体可定义如下:

$$\text{Entity} ::= \langle \text{个体} \rangle | \langle \text{构件} \rangle | \langle \text{过程控制的对象} \rangle \quad (5)$$

$$\text{Composite-activity} ::= \langle \text{Activity} \rangle | \langle \text{Activity} \rangle \\ \langle \text{Composite-activity} \rangle \quad (6)$$

$$\text{Complex-activity} ::= \langle \text{Activity} \rangle | \langle \text{Activity} \rangle \\ \Delta \langle \text{Complex-activity} \rangle \quad (7)$$

说明:式(5)说明实体可以是个体,或构件或过程控制的对象,式(6)和式(7)说明一个组合活动可以由多个简单活动组成,这个概念支持了在任何抽象级上模型化软件过程,例如一个高一级的活动可以由多个低一级的活动组合而成,式(6)说明低一级的多个活动之间可以是串行关系(串行组合),式(7)说明低一级的多个活动之间是并行关系(并行组合)。

一个串行组合活动的前置条件是序列中第一个活动的前置条件,后置条件是序列中最后一个活动的后置条件与序列中任何未受影响的后置条件的并运算。例如,一个串行组合活动 b 由 a1, a2, a3 三个简单活动组成,它们的前后置条件分别为:

$$(P1), a1, (Q1), (P2), a2, (Q2), (P3), a3, (Q3)$$

若假设 Q1 为: $(X1 > X2) \wedge (X3 > X4)$ Q2 为: $(X1$

$= X2) \wedge (X3 > X4)$, 则 b 的前置条件是 P1, b 的后置条件为: $Q3 \vee (X3 > X4)$ 。其中 $X3 > X4$ 是未受影响的序列中的后置条件。

一个并行组合活动的前置条件是序列中每个活动前置条件的并运算,后置条件是序列中每个活动后置条件的并运算,以上例的情况,若 b 为并行组合活动,则:

$$\text{b 的前置条件为: } (P1) \vee (P2) \vee (P3)$$

$$\text{b 的后置条件为: } (Q1) \vee (Q2) \vee (Q3)$$

式(7)中的并行符号“ Δ ”表示活动之间的并行关系,例如,假设活动 A1 由二个串行活动 a, b 组成, A2 由二个串行活动 c, d 组成,则:

$$A1 \Delta A2 = (abcd, acbd, acdb, cabd, cadb, cdab)$$

上式集合中的各元素均符合 a, b 串行, c, d 串行; A1 和 A2 并行的条件,即 a, b, c, d 中 a 在 b 之前, c 在 d 之前,其它任意排列。

3 软件过程模型的特点

式(1)~式(7)就组成了软件过程模型的形式化描述,该模型的特点如下:

1) 组合活动(串行组合和并行组合)的引入,使我们可以在任意的抽象级上描述软件过程,亦即引入了一种层次化的描述机制。

2) 活动是可以反复进行的,克服了瀑布模型的缺点,符合认识论的基本原理。

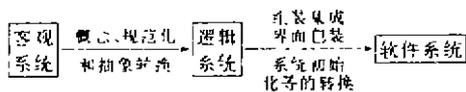
3) 把软件过程描述成一些可串行或并行的活动,可以使人们把注意力放在软件开发过程的执行上,而不是放在最终软件产品的状态上,这有益于实现软件工程的最终目标,把软件开发做为一种生产制造活动,而软件将是这种生产制造活动的产品,整个软件的生产制造过程将使用工程原理来管理和指导,就象生产制造汽车的生产线一样来自动化地生产制造软件产品。

4) 消息传递机制提供了人员之间协作,活动之间通讯的一个手段,也可以做为一个活动已完成的里程碑,这为项目管理提供了方便。

5) 由于前后置条件可以表示软件过程的各种属性和软件活动的各种联系,而这种属性和联系可以是文档等,这就为软件开发的管理和控制提供了一定的依据。

4 MIS 软件开发的过成模型

MIS 软件的开发经历了以下三个过程:



MIS 软件开发中三个系统之间的转换

我们可以使用前面论述的软件过程模型 SPM 来描述 MIS 软件开发的过程模型 SPM-MIS。MIS 软件开发的过程模型为以下三个活动：

- 建立客观系统模型(ERM)
- 建立逻辑系统模型(ELM)
- 建立软件系统模型(ESM)

下面符号“*”表示活动是可以反复的。

$$SPM-MIS = (ERM, ELM, ESM) \quad (8)$$

$$SPM-MIS = ((ERM)^*(ELM)^*)^*(ESM)^* \quad (9)$$

$$ERM ::= \langle \text{填表} \rangle \langle \text{标识个体等} \rangle \langle \text{分析验证} \rangle \quad (10)$$

$$\langle \text{填表} \rangle ::= \langle \text{业务部门1填表} \rangle \Delta \dots \Delta \langle \text{业务部门 N 填表} \rangle \quad (11)$$

$$\langle \text{标识个体等} \rangle ::= \langle \text{标识业务部门1} \rangle \Delta \dots \Delta \langle \text{标识业务部门 N} \rangle \quad (12)$$

$$\langle \text{标识业务部门 } i \rangle ::= \langle \text{标识个体} \rangle \langle \text{标识个体的动作和动作序列} \rangle \langle \text{标识个体间的关系} \rangle \langle \text{用一定格式的文档记录标识的结果} \rangle (1 \leq i \leq n) \quad (13)$$

$$\langle \text{分析验证} \rangle ::= \langle \text{分析验证业务部门1} \rangle \Delta \dots \Delta \langle \text{分析验证业务部门 N} \rangle \quad (14)$$

$$ELM ::= \langle \text{转换} \rangle \langle \text{用 CCDL 描述构件类} \rangle \langle \text{用 COL 操作构件} \rangle \quad (15)$$

$$\langle \text{转换} \rangle ::= \langle \text{个体1转换成构件类1} \rangle \Delta \dots \Delta \langle \text{个体 N 转换成构件类 N} \rangle \quad (16)$$

$$\langle \text{用 CCDL 描述构件类} \rangle ::= \langle \text{构件类1的描述} \rangle \Delta \dots \Delta \langle \text{构件类 N 的描述} \rangle \quad (17)$$

$$\langle \text{用 COL 操作构件} \rangle ::= \langle \text{操作构件1} \rangle \Delta \dots \Delta \langle \text{操作构件 N} \rangle \quad (18)$$

$$ESM ::= \langle \text{标识软件系统名称} \rangle \Delta \langle \text{标识版本号} \rangle \Delta \langle \text{查询选择构件} \rangle \Delta \langle \text{定义组装关系} \rangle \Delta \langle \text{用户界面包装} \rangle \quad (19)$$

$$\langle \text{定义组装关系} \rangle ::= \langle \text{指明查询路径} \rangle \Delta \langle \text{说明包含关系} \rangle \Delta \langle \text{指定命令行参数} \rangle \Delta \langle \text{给出构件通讯 FORM} \rangle \Delta \langle \text{设置环境变量} \rangle \quad (20)$$

$$\langle \text{用户界面包装} \rangle ::= \langle \text{设计菜单结构} \rangle \Delta \langle \text{描述 FORM} \rangle \quad (21)$$

式(8)说明 MIS 软件过程模型由三个活动组成,即建立客观系统模型、建立逻辑系统模型和建立软件系统模型;

式(9)说明这三个活动每一个都有可能反复,并且每个活动之间也可以反复,活动的反复符合认识论中人的认知过程;

式(10)说明 ERM 活动由三个串行的动作组成,即填表,标识个体等和验证;

式(11)、(12)、(14)说明 ERM 的三个动作根据业务部门,又可由多个并行的动作组成;

式(13)说明标识个体活动可分为四个串行的动作;

式(15)说明 ELM 活动由三个串行的动作组成,即转换,用 CCDL 描述构件类,用 COL 操作构件;

式(16)、(17)、(18)说明 ELM 的三个串行的动作又可分别由多个并行的子动作组成;

式(19)、(20)、(21)说明 ESM 活动由五个并行的动作组成。其中,定义组装关系的动作又包含了五个子动作,用户界面包装的动作包含二个子动作。

为了具体完成式(8)所述的三个活动,我们提供了三个支持语言,它们是,用结构化的自然语言来描述如何建立客观系统模型,用构件类定义语言和构件操作语言来描述如何建立逻辑系统模型,用过程控制语言来描述如何建立软件系统模型,有关这三个语言的具体形式和相应的支撑平台,我们将另文介绍。

结束语 本文提出的 MIS 软件开发的过程模型,已在一个实际 MIS 项目中进行了应用,该项目是财政部的八·五科研项目,目前其中的一个子系统已在试点单位投入运行,受到用户的欢迎,并正在向其它二个应用单位推广。

参考文献

- [1] L. Osterweil, Software Process are Software Too, Proc. of the Ninth Intl. Conf. on Software Eng., Monterey, California, April 1987
- [2] B. Curtis et al., Process Modeling, CACM, 35(9)1992
- [3] 仲萃豪等, 应用软件开发方法, 计算机科学, 1991, 1
- [4] 张然等, 软件过程的 JMOSP 模型, 软件学报, 1994, 10