

轨道模型(OM):大中型软件项目开发新模式的建构

邓冰 钟莹 陈春梨

(上海交通大学管理学院 上海200052) (复旦大学软件学院软件工程研发中心 上海200433)

摘要 轨道模型(Orbit Model)作为一种新的功能驱动、架构驱动和风险驱动的开发模式,可用于对复杂大中型软件项目进行流程管理,并已在项目开发中试行近两年。本文阐述了轨道模型的架构原理和方法,并希望引起业界对该新模式的探讨和实践。

关键词 轨道模型(Orbit Model,OM),软件架构,工程模式,流程管理,项目管理,能力成熟度模型(CMM)

Orbit Model: the Construction of a New Model for Middle & Large-Sized Software Projects

DENG Bing ZHONG Ying CHEN Chun-Li

(Management School of Shanghai Jiao Tong University, Shanghai 200052)

(The Software Engineering R&D Center of Software School of Fudan University, Shanghai 200433)

Abstract This paper delivers a totally new process model, Orbit Model, which has been applied for middle and large-sized software projects for past two years. The paper analyzes issues such as the basic construction of the new model and its application in practice in order to call for further discussions in it.

Keywords Orbit Model (OM), Software architecture, Engineering paradigms, Process management, Project management, Capability maturity model (CMM)

在软件项目的开发过程中,目前业界常用的开发流程模型(models)和模式(paradigms)有十多种,包括:

1. 线性或瀑布模式 (Linear Sequential / Waterfall Model);
2. 原型模式(Prototyping Model);
3. 快速开发模式(RAD / RP Model);
4. 增量或迭代模式(Incremental / Iterative Model);
5. 螺旋模式(Spiral Model);
6. 组件集合模式(Component Assembly Model);
7. 并行开发模式(Concurrent Development Model)^[1];
8. 形式或净室模式 (Formal Methods / Cleanroom Model);
9. 第四代语言模式(4GT)^[2];
10. 混沌模式(Chaos Model)等^[3]。

其中第1、3等模式较多用于中小型项目,4、5、6等主要用于中大型项目的开发。实践中,由于瀑布和RAD等模式不能满足较大项目对模块组织和架构规划的复杂要求,人们纷纷采纳以螺旋和增量/迭代模式为代表的新一代流程控制方法,Rational公司的统一软件过程(RUP)本质上属该范畴^[4]。本文探讨的轨道模型(Orbit Model,以下简称OM)由复旦大学软件工程研究与开发中心邓冰博士领导的工程应用模型研究组提出设想和原型,并在汇集了Apex、IBM、美标、上海泛思等公司的模型实施案例和应用数据后定义完善(详细应用数据另见论文“OM模型实施数据分析与建模方法评估”)。该模型试图协调一般中大型项目开发中易引发冲突和失败的要素,如架构偏差与风险失控的相关性等。本文对OM模型的架构原理和方法进行了阐释。

1. 模型定义

从项目生命周期的角度,OM模型把一个完整的软件项目分为七个核心流程,即:需求(Requirement);分析(Analysis);架构设计(Architecture Design);实施设计(Implementation Design);实施及复查(Implementation & Review);测试(Testing);维护及升级(Maintenance & Improvement)。每一核心流程都可用一个OM模型视图来表示,如图1所示。每一个核心流程又可划分为四个阶段:分析(Analysis);设计(Design);实现(Implementation);评估(Assessment),即ADIA阶段。

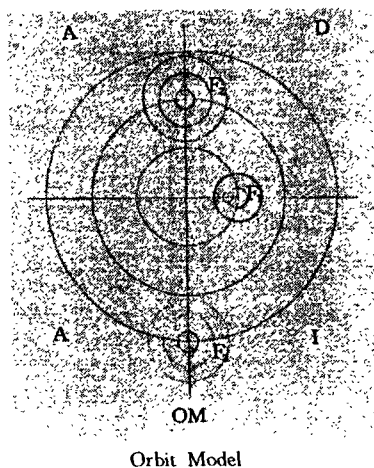


图1 轨道模型基本形态

邓冰 留美博士,复旦大学软件学院软件工程研发中心常务副主任,上海交通大学管理学院及新加坡南洋理工大学 MIS/ERP 教学,美国智能研究院有限公司主咨询师 CCO。钟莹 MSE,上海科旭通信电子有限公司 PM/SQA。陈春梨 BS,复旦大学并行处理研究所。

2. 模型定义2

OM 模型采纳了功能驱动式的建构。从第一个核心流程，即需求流程的启动开始，OM 要求将项目/产品所应实现的主要的最终业务功能(Final Business Functions)作为启动功能(Initiative Functions)，来引导整个模型的建构和开发流程的实现。以项目 P 为例，设定客户要求 P 最终实现的主要业务功能有3项，分别为 F1, F2, F3, 那么在需求流程中我们就有了3个模块(见模型定义3)，分别代表3项启动功能。这3个模块按顺时针方向将分别经历需求分析、需求设计、需求实现和需求评估等四个 ADIA 阶段。

3. 模型定义3

从架构的角度，OM 模型将项目切分为四个依次包容的组合层次(layers): 1. 模块层(Module Layer); 2. 单元层(Unit Layer); 3. 套件层(Package Layer); 4. 组件层(Component / Class Layer)。模块和单元层主要用于整个项目架构的规划与设立，而套件与组件层则用于在此架构上完成项目的具体开发。OM 模型中所谓的轨道就是一组贯穿于每一层架构，并引导架构和流程实现的业务功能的规则。实践中我们发现，这组规则不仅从业务功能上定义了各层架构，而且还与架构开发各组成部件的技术/流程定义密不可分。我们把 OM 模型中的轨道正式定义为：“业务功能规则或逻辑图，同时起着流程引导与沟通的作用”。从图1中可以看到项目 P 的3个模块 F1, F2, F3 分别由3层轨道所引导。模型中轨道的数目与模块数一致。

4. 模型定义4

确立启动功能之后，OM 模型将按照定义1中的核心流程顺序发展到分析和架构设计流程，并完成对项目全部模块层和单元层的设计与建构。OM 模型的建构原则包括功能驱动、架构驱动和风险驱动三大特征，因此分析和架构设计流程是它的主要工作流程。如图2所示。

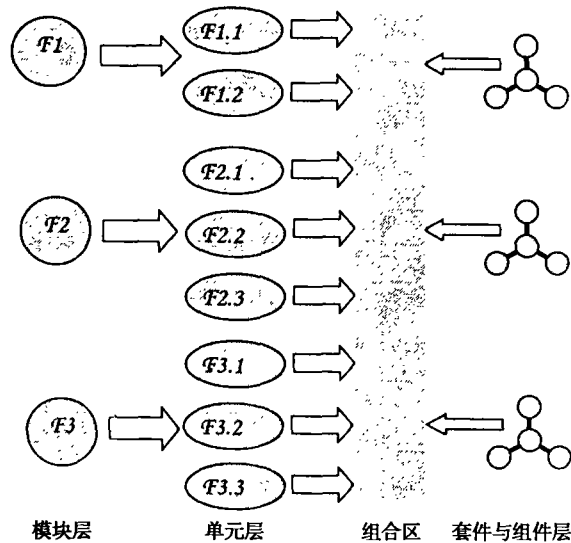


图2 轨道模型架构展开形态

图2是 OM 模型基本形态的架构展开示意图。项目 P 的3

个模块 F1, F2, F3 在分析和架构设计的流程中被进一步细分为单元层面的8个单元，其中 F1 包含2个单元，F2和 F3 各含3个单元。图中向右方的横向箭头表示模块层面的轨道被细分为单元层面的轨道，由此，业务功能规则保持了持续性，各层及各元素之间的信息也可因此沟通。

5. 模型定义5

各层的轨道(业务功能规则)通常应在模块、单元等构件之前设立。由于需求等因素的常变性，业务功能会经常性的变动。因此，定义中的建构顺序可能出现变异。例如某些模块和单元的确立可能会有多次反复、修正，甚至推迟至流程后期才能明确；某些组件和套件的开发可能会打破正常次序。作为一个面向中大型复杂项目的综合模型，OM 模型的运用应该是可裁剪和柔性的，只是这类变更必须置于流程控制之下^[5]。

6. 模型定义6

OM 模型的另一独特之处是将需求、分析和架构设计流程与后四项流程，如实施设计等明确分开。需求等前三项流程主要解决项目的整体架构，而后四项流程主要解决项目的具体实施。与大多数开发模式不同，从架构设计之后的实施设计开始，OM 模型不在已有的单元层进行全面的架构细化实施，而是在单元层的整体框架内，选择一或数个最核心的底层组件进行实施，使其迭代成为套件后逐步向单元层靠拢，最终在组合区完成整个项目的组合与建设。如图2中左向箭头所示。这种双向建构的主要意义在于：既保证了整体架构的正确性(与最终业务功能的吻合)，又可避免整体开发失败的高风险。在模型定义6的具体实施中，我们推荐基于组件的面向对象技术(Component-based OOP)和基于组件的面向主体技术(Agent-Oriented Programming, AOP)等方法。按照 OM 模型的要求，构成组件层轨道的业务功能规则应该引导所有组件的具体实施设计，实施和复查等，它同时能为组件中的对象以及组件层之间的沟通提供消息传递机制。有关组件及主体技术在中大型系统开发中的应用，请参考复旦软件工程研发中心 Agents 课题组的论文。

上述六项定义是 OM 模型的基本架构定义。其它相关建设还包含 OM 流程规则，OM 模板和 OM 评估表。篇幅所限，不在此一一赘述。

参考文献

- 1 Humphrey W S. Software Process Modeling: Principles of Entity Process Models. In: Proc. 11th Intl. Conf. on Software Engineering, The U. S. A. : IEEE Computer Society Press, 1989. 333~337
- 2 Pressman R S. Software Engineering A Practitioner's Approach. Version 4. The U. S. A. McGraw-Hill Companies, Inc. , 1997. 23~47
- 3 Raccoon L B. The Chaos Model and the Chaos Life Cycle. ACM Software Engineering Notes, 1995, 20: 61~65
- 4 Booch G, Jacobson I, Rumbaugh J. The Unified Software Development Process. The U. S. A. Addison Wesley Longman, Inc. , 1999. 5~79
- 5 Humphrey W S. A Discipline For Software Engineering. The U. S. A. Addison-Wesley, Inc. , 1995. 309~437