

基于角色模型的业务过程再工程(BPR)的研究

徐焕良^{1,2} 李绪蓉¹ 丁秋林¹

(南京航空航天大学 南京210016)¹ (江苏省盐城工学院 江苏盐城224002)²

The Study of Business Process Reengineering (BPR) Based on Role Model

XU Huan-Liang LI Xu-Rong DING Qiou-Lin

(Nanjing University of Aeronautics and Astronautics, Nanjing 210016)¹

(Yancheng College of Technology, Yancheng 224002)²

Abstract Based on role object, the Business Process Reengineering (BPR) is quickly implemented by constructing the frame of role model. This technology will be widely applied in CIMS too.

Keywords Role model, Frame, BPR

1 概述

随着信息技术的高速发展,特别是 Internet 应用的普及,人类的思维习惯和行为方式已发生了根本性的改变。在制造业,新技术的诞生和应用,使得并行工程、敏捷制造和虚拟制

造等先进理念得以实现。传统的企业科层式(树型)的组织结构亦已逐渐扁平化,并从面向功能的组织转向面向过程的组织。组织结构的变化,也使人们重新审视企业经营过程。为求得在成本、速度、质量等方面的显著提高,大量的业务过程的重组(BPR)作业开始实施。

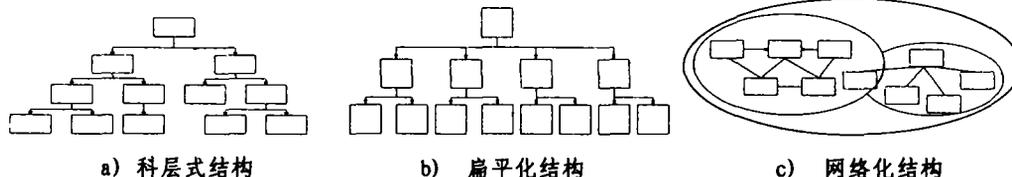


图1 企业组织结构的演变

BPR 旨在企业组织中及组织间的工作流程与程序进行分析和再设计(Davenport & Short 语)。众所周知,在传统科层式体系结构中,由于高度专业化的分工,个体之间各自孤立,组织结构呈现出刚性特征。这种刚性的组织结构,由于在组织中及组织间的个体形成了一个“信息孤岛”,难以、甚至无法进行信息的交互。即使发生通讯,其信息传输的复杂、交错和冗余,也极大地制约了业务过程流畅、运转的有序。特别是,由于外部市场需求发生变化,或客户个性化服务所派生的不同要求而必须进行业务过程重组或优化时,它就无法适应、更无法进行调整以支持过程的重组。在扁平化的结构里,网络化、面向过程的管理,信息能够迅速地交互,过程可以及时地控制。特别 BPR 实施时,常常涉及到与业务活动相关联的业务部门的调整问题,本文基于“角色(Role)”处理来减少易变的业务部门(的组织方式)与相对稳定的业务活动角色这两种因素之间的耦合性,使其可以极具柔性地响应和支持业务过程的变化,快速地实现业务过程的再工程。

2 面向过程的工作流管理模式

面向过程的管理模式是基于这样一个概念:任何一项复杂的业务项目(或称任务、生产、业务等。比方,一个承接的项目、一个生产的任务等)都可以分解成一系列相互关联的而又相互独立的串行或并行的子任务。我们可以表示为:

$$\text{Task} = \{T_1, T_2, \dots, T_n\}$$

而每个子任务根据性质又可以分为许多活动步骤。这些活动总是向着希望的状态和方向发展。亦可表示为:

$$\text{Activity} = \{A_1, A_2, \dots, A_m\}$$

每个活动根据性质(需求)又有许多成员角色实施:

$$\text{Role} = \{R_1, R_2, \dots, R_k\}$$

以上三个要素,支持项目的实施和完成。当然,为能够使项目或任务有序、高效地实施,还需制定出任务、活动、角色及控制、协调策略。即确定各任务之间、活动之间、角色之间、任务与活动之间、活动与角色之间的关系。

$$\text{Strategy} = \{S_1, S_2, \dots, S_l\}$$

这样,我们可以这样来表述其项目的工作系统:

$$\text{Item} = \{T, A, R, S\}$$

表示该系统有任务 task、活动 activity、角色 role,并遵循 strategy 所确定的规则。

于是,一个项目或任务的执行过程就可以分为过程定义、任务分解、确定活动、角色配置、交互控制和结果输出,如图2所示。

企业里的任何一个项目或任务我们可以分解成一系列有序的活动。当然这些活动总是向着用户希望的方向或目标运行。操作这些活动的成员,我们定义称为个体角色或角色。角色是具有一定性质、或者说能够完成某种使命、具有某种功能

徐焕良 副教授,博士生,主要研究方向:CAD/CAM/MIS/CIMS、企业信息化、电子商务等。李绪蓉 博士生,主要研究方向:CAD/CAM/MIS/CIMS、企业信息化、电子商务等。丁秋林 教授,博导,主要研究方向:CAD/CAM/MIS/CIMS、企业信息化、电子商务等。

的对象。它可以是人、组织机构或者更广泛意义上的设备等对象体。显然,一个活动可能对应多个角色,不同的对象可能使活动呈现不同的形态、或者性质,即,角色的性质将决定活动的性质。而一个角色也可能同时支持或参与几个活动的操作,但什么样的活动决定着需要什么样的角色,亦即,活动的性质也决定着角色的性质(图3示)。当然,一个活动必然有一个最优的角色群来完成,而一个角色也必然有一个最优的合适的活动。

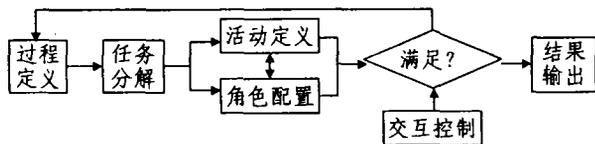


图2 工作流程图

因此,活动与角色的关系为:每个活动可以配置许多角色,每个角色也可以参与许多活动。活动的性质决定选择和配置角色性质。角色的性质也可以决定活动的性质。活动与角色的相互作用,使得业务过程发生变化时,可以通过调整活动项,相应重组成员角色来响应和支持活动的调整;或者调整成员角色,改变活动的发展方向,从而重组或优化业务过程。这就是说,在业务执行过程中,可以基于角色模型对业务过程进行优化或重组。

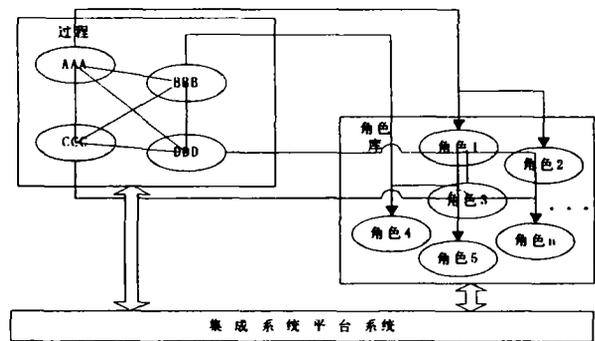


图3 基于角色模型管理结构

交互控制的作用,是提供一个机制。强调任务执行过程中的有效控制。在业务过程中,都存在着事实上的带有特征性的关键点(或称项目里程碑),当关键点输出结果显示与设计状态发生异常,就需要调整其业务过程,可能是重新定义流程,重新分解任务,并根据某项规则,重新制定活动流转,同时配置相应的角色,以及时地响应和支持这种变化,完成业务过程快速重构。其运算为有限循环迭代直至正确输出。

3 业务过程形式化表示

在业务实例的执行过程中,我们分别用定义活动(Activity)实例、角色(Role)实例及业务过程实例的形式化表示,以说明业务过程的运行和基于角色模型管理的业务过程重构方式。

3.1 业务过程实例的形式化表示

过程实例 P 某一业务项目的执行过程。

$$P = \{n, n', b, e, s, A, R\}$$

其中, n 表示过程实例的名称; n' 表示相应过程定义的名称; b 表示 P 的开始; e 表示 P 的结束; s 表示过程实例当前运行状态,可以是正在运行、被禁止、挂起或完成; A 表示自开始以来

所有活动实例构成的集合; R 表示自开始以来所有参与的角色实例构成的集合。

需注意的是, A、R 是描述整个过程的实例状态。管理人员可以使用管理工具对过程实例的状态加以人为的控制,如暂停、或恢复某过程实例的运行。

3.2 活动实例的形式化表示

活动实例是业务过程实例的一个步骤。我们用

$$a = \{n_a, R_a, R'_a, b_a, e_a, s_a, V_a\}$$

其中, n_a 表示与该活动实例相应的业务过程名称; R_a 表示参与活动中的所有角色名称构成的集合; R'_a 表示在前驱活动的分量; b_a 表示 A 的开始; e_a 表示 A 的结束; s_a 表示 a 的当前状态,准备、准备好、正在运行、完成。一般而言,活动实例状态是由业务过程实例运行情况而维护的; V_a 表示被处理的所有命名值构成的集合,表示此活动实例中需处理的各种数据。

3.3 角色实例的形式化表示

角色实例是参与活动的成员实例。

$$r = \{n_r, A_r, A'_r, b_r, e_r, s_r, V_r\}$$

其中, n_r 表示与该活动实例相应的业务过程名称; A_r 表示角色所参与的所有活动的集合; A'_r 表示当前前驱的活动分量; b_r 表示 R 的开始; e_r 表示 R 的结束; s_r 表示 r 的当前状态,准备、准备好、正在运行、完成。一般而言,角色实例状态是由业务过程实例或活动实例运行情况而维护的; V_r 表示被处理的所有命名值构成的集合,表示此角色实例中需处理的各种资料。

3.4 算法

- 1) 项目开始。启动某一业务过程 name
- 2) 根据用户指定的过程实例名称 name, 建立一个新的过程实例结构 p, 并使
 $n' \leftarrow n_p; b_p \leftarrow$ 系统当前时间;
 $n \leftarrow name, A, R \leftarrow \Phi;$
- 3) 根据起始活动 a₁, 建立一个新的活动实例 a, 并使:
 $n_a \leftarrow n_{a1}; R_a \leftarrow \{r\}; R'_a \leftarrow \Phi;$
 $b_a \leftarrow b_p; V_a \leftarrow \Phi; a' \leftarrow$ 准备好
- 4) 或者根据初始角色 r₁, 建立一个相应的角色实例 r, 并使:
 $n_r \leftarrow n_{r1}; A_r \leftarrow \{a\}; A'_r \leftarrow \Phi$
 $b_r \leftarrow b_p; V_r \leftarrow \Phi; r' \leftarrow$ 准备好
- 5) P ← {p}

至此,完成一个过程实例的创建,同时相应建立了活动实例或角色实例。需要说明的是,建立的活动实例中,包含了参与此活动的所有角色的集合。而建立的角色实例中,也反映出该角色所参与的所有活动的集合。

6) 活动/角色的移交

活动 a 表示业务过程中一个实际或抽象的工作步骤。A、R、表示从业务过程抽象出来的所有活动和角色的集合。在业务过程运行中,每一个过程确定着唯一的“起始活动”或“起始角色”。我们用一个四元组来表示:

$$P = \{n, A, R, F\}$$

其中, n 表示业务过程的名称; A 表示所有活动集合; R 表示所有角色集合; F = A × A × C, 或 F = R × R × C。(注:如果业务过程以起始角色开始,并以角色移交时,运用该式计算)它描述 A 中各活动之间的关系及 R 中角色之间的关系。

C 中的每一个元素 c 是一个二元组 {c, E}, 其中 c 为布尔表达式, E 为多个命名表达式构成的集合。C 描述了在条件 c

的计算结果为真,激活后续的哪一个活动项或角色并把数据传递给它。例如,对任一 $f=(a_1, a_2, c) \in F$, 或者 $f=(r_1, r_2, c) \in F$, 并且 c 计算为真,那么 $V \leftarrow E$ 将被传送给 a_2 或 r_2 处理。

这种移交或称连接(Connect),可分为定义活动执行顺序的控制连接和表示信息关联性的数据连接,如上所描述的 c 和 $V \leftarrow E$ 。一个活动可以发出多个连接,也可以接受多个连接,每个连接可以定义附加性的转移条件。连接关系的种类可概括为以下几种形式:

```

Connect
Rating-of-connect: INTEGER
target-rating: INTEGER
description: TEXT
State: Boolean // 状态,“是否被激活”
issuer: POINTER // 指向 Participant-member 类的指针
receiver: POINTER // 指向 Participant-member 类的指针
transfer() // 连接的转移
Choose CASE Type {
    Triggers(); // 触发
    AddValueTo(); // 为...增值
    ConsistOf(); // 由...构成
    AimsTo(); // 以...为目的
    InterfaceTo(); // 与...关联
    ExplodesTo(); // 分解为
    Governs(); // 主导/控制
    Adduce(); // 调用
    ExcutedBy(); // 由...执行
}
END CASE
End Connect
    
```

需注意的是,上述移交蕴涵着活动和角色的两种方式。当上一个活动完成后当然地会转送到下一个活动,而当角色的所处理的数据提供时,也就可以立即转送下一个角色处理,其中的活动实例中角色前驱分量 R_i 、角色实例中的活动前驱分量 A_i ,决定着活动和角色的性质。这为业务过程能够合理再重构提供了保证。

4 基于角色模型组件的 BPR 实现机制

4.1 角色模型

角色模型是描述角色对象及其相互间的协调行为。一个角色描述一个对象在系统行为中所承担的责任。角色对象描述一个或一个以上的责任。角色模型抽象地描述对象在系统

协调行为中的作用,得以建立角色间的协调、继承、使用机制。图4中表示了一个角色对象 role 与一个或一个以上的角色对象的协调关联。图中协调、继承和使用分别用细实线、带尖头的细实线和带黑尖头细实线表示。角色约束的描述一般用自然语言描述。随着角色的细化,也可使用 OMG/OCL 描述。

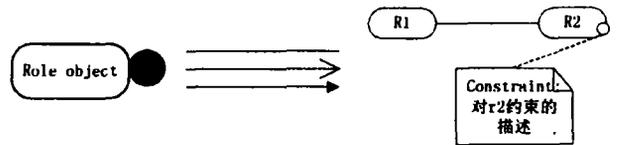


图4 RM的基本图元素

4.2 基于角色模型组件的 BPR 实现

4.2.1 基于前驱分量 R_i, A_i 的业务规则确定 参照图4角色模型的基本图元素,在业务过程中,某一活动所对应的多个角色对象的使用关系、活动之间的继承关系、角色对象之间的协调关系,我们以图5表示。

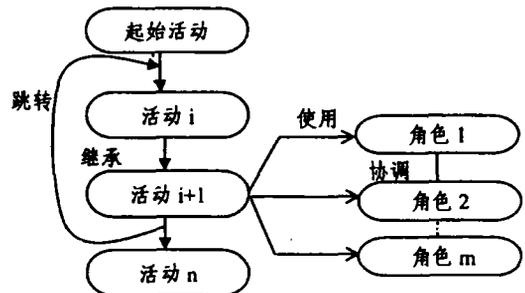


图5 基于角色模型的活动、角色、活动与角色关系表示图

在面向对象的分析方法中,一个业务过程可以被理解为一组对象间的一系列协同的请求/服务操作。对象的继承语义和多态行为可以大大增强系统的可扩展性和可适应性;服务以对象操作的方式来实现,也易于实现系统职责的分配;对象间的协同、冲突机制,一般地,是基于蕴涵在业务过程中业务规则进行操作。图6表示基于规则的业务活动流转。

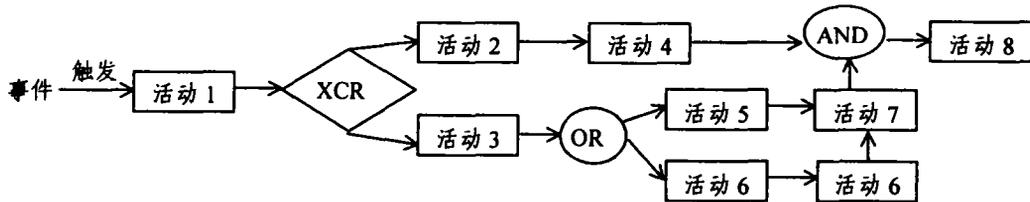


图6 基于规则的业务过程中活动的流转

```

基于规则的描述:
BusinessRule
Event-Table: POINTER // 指针型(指向事件列表)
Related-Object: POINTER
CurrentRule // 执行当前的有效规则
{ rule-name: TEXT
  rule-description: TEXT
  rule-specification: TEXT
  next-task {
    task-type: (action, rule) // 规则引导活动
    task-name: TEXT
  }
}
END CurrentRule
IfElseRULE // 转而执行另一个有效的规则
{ rule-name: TEXT
  rule-description: TEXT
  rule-specification: type {}
condition { condition-description: TEXT
  condition-specification: RuleType()
}
}
    
```

```

then-task { task-type: (action, rule)
  task-name: TEXT }
elseTask {
  task-type: (action, rule)
  task-name: TEXT }
}
Choose CASE Rule-type { // 规则对象的类型
  Function-Rule(); // 功能规则对象
  Behavior-Rule(); // 行为规则对象
  Structure-Rule(); // 结构规则对象
}
END CASE
endifElseRULE
    
```

不失一般性,我们用 P 表示过程, P_i 表示某一过程的操作。那么,过程操作之间的约束包含有 And-Join, And-Split, Or-Join, Or-Split 等形式,其表示方法如图7所示(图中的 P_A, P_B, P_C 分别表示操作 A、操作 B、操作 C)。And-Join 表示激活操作 C 的前提条件是操作 A 及 B 已经完成;Or-Join 表示激

活操作 C 的前提条件是操作 A 或 B 已经完成;And-Split 表示完成操作 A 后,可以激活操作 B 和 C;Or-Split 表示完成操作 A 后,可以同时激活操作 B 或 C。

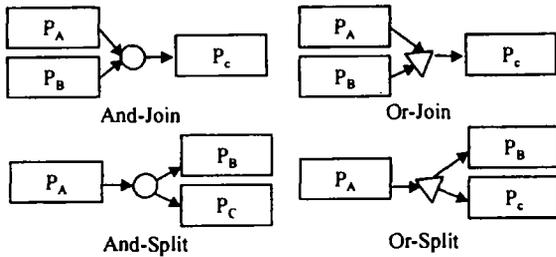


图7 操作之间约束的表示方法

需注意的是,各种操作的协同,以及规则的确定,是基于

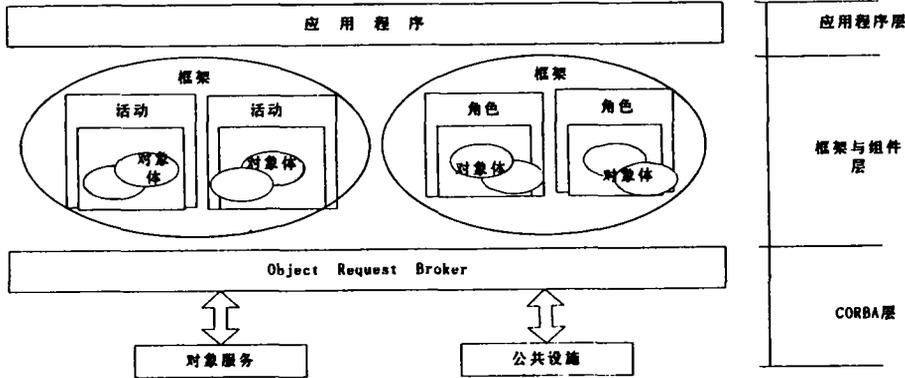


图8 基于角色模型组件的框架结构体系

这样,在某业务实施过程中,基于角色模型组件框架体系,以支持 BPR 实施.通过业务规则,触发活动跳转,快速配置角色,来完成最终完成业务过程的重组(图8示)。

结束语 BPR 的理念,表述了“对经营过程的彻底的重新构思、根本的重新设计以达到在一些诸如成本、质量和速度等关键性能方面的显著提高”(海默·潜培语)。本文试图从更广泛的意义上抽象出角色模型的概念,根据业务过程中的一系列逻辑相关的活动步骤,通过角色有效地执行它们以达到预定的业务结果.特别是在业务过程变化时,能够基于角色模型,快速实现重组.例如,对一个生产型项目,活动实施角色可能是人员、组织,甚至设备(如各种机床设备)。当加工工艺过程需要变化时,就可以通过角色模型—机床设备转换,实现加工工艺过程的重组.这在 CIMS 系统中,无疑极具现实意义。

在活动实例中的角色前驱分量 R_i 和角色实例中活动前驱分量 A_i 制定的.在称之为里程碑的 Key Point 上的 R_i 、 A_i 的状态是发生活动跳转的触发源.亦即,当关键点 R_i 、 A_i 输出结果显示与设计状态发生异常,则转而执行另一个有效规则,来实现活动跳转.同时,根据活动的性质,配置新的角色,从而迅速实现业务过程的再工程(参照规则的 PDL 的描述及图2表示的工作流程图)。

4.2.2 基于角色模型的 BPR 实现 一个业务过程,其活动和角色是相对稳定的对象,而业务规则,随着不同的业务项目,多变的市场状况而呈现出不同.为此,我们把表现实体或过程及其相互关系的相对稳定部分(比方活动、角色及其它它们之间的相互关系)与表现为规则及策略方法指导下的动态行为的易变部分(比方业务规则)进行剥离,构建基于角色模型的组件框架,如活动组件、角色组件,如图8所示。

当然,在这方面的还有许多亟待研究的工作,但有一点是肯定的,那就是利用现代信息技术,基于角色模型的概念,重新设计我们的业务过程,完全能够达到诸如成本、质量和速度等方面的极大提高。

参考文献

- 1 余菁编著.企业再造:重组企业的业务流程.广东经济出版社,2001.10
- 2 何克清,等.角色模型化:—一个软件模式知识级的建模方法.计算机科学,2001,28(8)
- 3 Object Management Group. Common Object Request Broker: Architecture and Specification, Feb. 1998
- 4 W3C Note 18 December 2001. <http://www.w3.org/TR/2001/NOTE-daml+oil-walkthru-20011218>
- 5 Connolly D, et al. DAML+OIL (March 2001) Reference Description, W3C Note 18 December 2001. <http://www.w3.org/TR/2001/NOTE-daml+oil-reference-20011218>
- 6 <http://www.w3c.org/XML/>
- 7 Thompson H S, et al. XML Schema Part 1: Structures, W3C Recommendation, 2 May 2001. <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>
- 8 <http://www.oasis-open.org/cover/schemas.html>
- 9 Bray T, Hollander D, Layman A. Namespaces in XML, World Wide Web Consortium 14-January-1999. <http://www.w3.org/TR/REC-xml-names>
- 10 <http://www.ontoknowledge.org/>

(上接第141页)

- 8 Lassila O, Swick R R. Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation 22 February 1999. <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>
- 9 Brickley D, Guha R, et al. Resource Description Framework (RDF) Schema Specification, W3C Candidate Recommendation 27 March 2000. <http://www.w3.org/TR/2000/CR-rdf-schema-20000327>
- 10 Horrocks I, et al. The Ontology Interchange Language OIL: [Technical Report]. Free University of Amsterdam, 2000. <http://www.ontoknowledge.org/oil/>
- 11 The DARPA Agent Markup Language Homepage. <http://www.daml.org/>
- 12 Connolly D, et al. Annotated DAML+OIL Ontology Markup,