

# 一种基于语法的供应链流程定义元模型<sup>①</sup>

严建援<sup>1</sup>, 李 凯<sup>2</sup>, 张 路<sup>1</sup>

(1. 南开大学商学院, 天津 300071; 2 南开大学泰达学院, 天津 300071)

**摘要:** 在供应链流程建模和描述的方法体系中, 语法建模方法是最新出现的一种很有发展潜力的建模语言, 与其它建模方法相比具有多方面的优越性. 供应链流程定义语言 (SCPDL, supply chain process definition language) 作为一种新的供应链流程建模方法, 是供应链流程与语法模型的转换器, 它可使供应链流程转化为相应的语法模型, 以便在系统上实现对流程的查询、管理等逻辑处理, 并进一步实现了不同方法描述的供应链流程之间的便利转换. 通过对流程定义语言的深入研究, 融入最新的过程语法建模和协调理论的相关理念, 提出了一个基于语法的供应链流程定义元模型. 并在此基础上采用 XML (eXtensible Markup Language) 对其规则进行重写, 形成了基于 XML 的供应链流程定义语言. 提出的 XSCPDL (XML SCPDL) 模型方法将使供应链管理的建模、管理和优化工具与供应链流程引擎之间实现相互转换, 并便于不同软件开发商提供的流程模型在 XML 的平台上进行快速交互.

**关键词:** 供应链; 元模型; 语法建模; 流程定义语言 (PDL); XML

**中图分类号:** 931.6      **文献标识码:** A      **文章编号:** 1007-9807(2010)06-0033-11

## 0 引 言

20 世纪 90 年代以来, 供应链管理逐渐成为企业获取竞争优势增强竞争力的重要手段. 为了更好地实施供应链管理, 研究供应链流程, 建立相应的供应链流程管理信息系统来实现供应链流程的管理和优化是十分必要的<sup>[1]</sup>. 在日益激烈的竞争当中, 业务流程的查询, 优化和重组已经成为供应链竞争的重要手段<sup>[2]</sup>. 各种支持业务流程重组 (BPR) 的建模工具和软件业正像雨后春笋般出现, 然而它们往往使用的标准各不相同, 也难以实现相互之间的交流和共享<sup>[3-4]</sup>. 在供应链流程管理中实现流程的查询、存储和优化之前, 有一个重要问题亟待解决, 那就是供应链流程定义语言的设计<sup>[5]</sup>.

在业务流程定义语言的研究方面, 工作流管

理协会 (work flow management coalition W FMC) 走在了前面. 工作流管理协会早在其成立的第二年 (1994) 就发布了工作流管理系统的参考模型, 并根据工作流的建模思想提出了早期的工作流定义语言 (workflow process definition language WPDL), 后来将这一标准变更为 XPDL<sup>[6]</sup>. XPDL 是至今工作流领域最为重要的一个标准, 目前大多数工作流引擎是依据该标准设计开发的. 不久之后, 由于 XPDL 在定义模型方面的局限性, BPM I 发布了 BPM L (business process model language) 规范, 与 W FMC 合作制定业务流程和工作流标准<sup>[7]</sup>. 同时应用比较广泛的还有 OMG 发布的 Work flow M anagement Facility 这个标准用于定义了如何将工作流向 CORBA 转换<sup>[8]</sup>. 随着 Web 的广泛使用以及电子商务的兴起, 最近还出现了一些新的基于工作流的流程转换标准, 比如

① 收稿日期: 2008-03-03; 修订日期: 2009-09-22

基金项目: 国家自然科学基金资助项目 (70471040); IBM SUR 资助项目.

作者简介: 严建援 (1955-), 女, 辽宁人, 博士, 教授, 博士生导师. Email: yan5559@126.com

WSCI<sup>②</sup>、ebXML<sup>③</sup>和BPEL<sup>④</sup>等。

## 1 基于语法的供应链流程建模方法

基于语法的供应链流程定义元模型是在语法建模方法的基础上提出的,因此有必要回顾一下语法建模和基于语法的供应链流程建模方法。

### 1.1 语法建模

基于语法的过程建模方法通过供应链过程分析来研究事件发生的序列,并从中发现有助于供应链过程表达与创新的规律,将供应链过程及语言的核心概念进行对比映射,根据一个有限的字典和一组有限的规则或约束来产生一组模式的集合,进而为供应链过程的再造提供一组解空间,以供应链上的各企业和伙伴进行决策<sup>[9]</sup>。

语法化方法最早出现在对语言学中的重要概念语法的研究中<sup>[12]</sup>。语法就是运用一个有限的字典和一组有限的规则或约束(rules or constraints)来描述一组模式的集合(可能是无限的),这些规则或约束详细说明了所允许的字典元素的组合<sup>[13]</sup>。语法可以描述所有可能模式的集合,同时提供一种可以解释这些集合的规则。在语言学中,“语法”被认为是“可繁殖的”(generative),它拥有从有限的字典和规则中生成无限句集的能力。同样,“可繁殖性”(generativity)同样是组织理论中的研究热点,因为这种特性可以让我们基于一套约束或约束的变化来预测新的组织过程和形式<sup>[14]</sup>。基于这种对应关系,Manbe等人在1993年提出,将语法的方法用于过程的建模,其语法模型的“可繁殖性”就可以为我们提供新的设计过程的办法<sup>[15]</sup>。

语法的方法多年前就曾用来与组织过程相联系。第一个将语法方法应用于组织理论的是Weick<sup>[11]</sup>,他在1979年将“组织形成过程”定义为一种“确认语法的过程”,并进一步提出“组织形成过程是一种语法,一套编码或一剂处方”。而Drazin和Sandelands<sup>[16]</sup>于1992年提出组织形成过

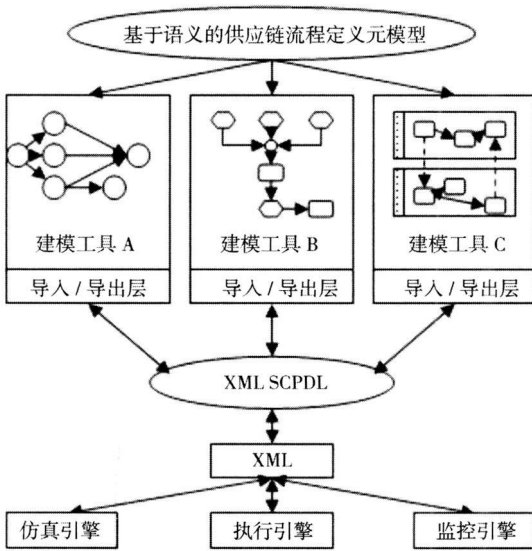


图 1 流程转换的过程

Fig. 1 Process transformation process

工作流建模是一个应用广泛且比较成熟的建模方法<sup>[9]</sup>,但是新出现的语法建模方法在可分解性、可繁殖性等各方面显示出了无可比拟的优越性,已被公认为是一种很有发展潜力的建模语言,特别是在供应链流程建模领域<sup>[10-11]</sup>。本研究将最新的过程语法建模方法和协调理论的相关理念融入供应链过程建模,创新性地构建了一套供应链过程语法建模和描述的方法,提出了一个基于语法的供应链流程定义元模型。这个基于语法的供应链流程定义元模型将为以后各种基于语法的建模软件工具提供一个基本标准,所有遵循元模型的建模工具所建立的供应链流程模型都将可以通过供应链流程定义语言进行转换(见图1),成为一种XML格式的标准数据形式,进而通过流程引擎实现流程的查询、仿真等。这样,实现了供应链管理的建模、管理和优化工具与供应链流程引擎之间的相互转换,同时使得由不同软件开发商提供的流程模型可以在XML的平台上进行交互。

② 2002年6月26日, BEA, Intalin, SAP, Sun四家公司提出了基于xml的WSCI规范,推动Web服务进入了一个全新的阶段。这个规范主要描述了一个参与和其它服务进行协作交互的Web服务所交换的消息流。

③ ebXML是一组支持模块化电子商务框架的规范。是联合国(UN/CEFACT,贸易促进和电子商务中心)和OASIS(结构化信息标准发展组织)共同倡导、全球参与开发和使用的规范。

④ 2002年8月9日, Microsoft, BEA, IBM, SAP & Siebel联合提交发布了BPEL规范。BPEL联合了一系列规范,比如XLANG, WSFL, B4ML

程的“深层结构”是由一种“可衍生的语法”构成的。通过用语法术语描述组织过程, 这些学者以及其他许多学者<sup>[17-22]</sup>都含蓄地指出, 组织也是有语法的, 就好像人类的语言一样。这就提供了一种对于组织形式和组织过程进行理论化的新方法, 它也同样可能为其他随时间变化的过程(例如: 组织生命周期、革新、战略形成)创造语法的模型。

美国麻省理工学院斯隆管理学院的协调科学研究中心(Center for Coordination Science, MIT Sloan School of Management)从 20 世纪 90 年代中期开始就致力于供应链协调机制及其过程优化工具方面的研究。1993 年 5 月 MIT 协调科学中心的 Malone 等人正式提出一种创建组织的工具——组织过程手册(A Handbook Of Organizational Processes)<sup>[15, 23]</sup>。过程手册收集了有着相似过程的众多组织案例, 并通过一种在线“过程手册”将这些案例表述出来, 同时指出各种过程的优缺点和备选方案。过程手册为语法化的建模方法奠定基础。1994 年 8 月美国加利福尼亚大学安德森管理学院的 Pentland 教授在前人研究的基础上, 将语法隐喻发展成一种描述组织工作过程的严格的理论模型, 正式的提出了一种组织过程的语法模型。其语法模型始于一个基本作业(或步骤)的字典, 并且指出这些作业如何组合而成为过程。与其他的数据库分析技术不同, 该语法的模型提出了一种能够表达组织过程所具有的嵌套和层次特性的有效办法<sup>[24]</sup>。他同时将语法和组织过程的进行了映射, 他将组织行动(moves)映射为单词, 将程序(performance program)映射为句法成分, 将过程(processes)映射为句子, 而组织和制度结构(organizational and institutional structure)则对应于重写规则(rewrite rules)。为语法化建模方法的后续研究奠定了基础。

## 1.2 基于语法的供应链流程建模方法

供应链流程是为达到一个特定目的的一组逻辑相关的有序的活动<sup>[25-26]</sup>。语法建模方法与其它方法相比在供应链流程建模上有很大的优越性。一般来说, 语法建模方法主要包括元素定义和语法规则。

### 1.2.1 元素定义

语法建模方法的基本语法元素可定义为以下

三种: 过程、事件、依赖<sup>[27]</sup>。供应链过程元素词典使供应链流程中的过程实现了标准化。将描述过程元素词典的元素可以定义为名为“过程”的对象。“过程”的属性被定义为

$$P_i = \langle Id(P_i), Name(P_i), Description(P_i), In(P_i), Out(P_i), Agent(P_i), Position(P_i), preD(P_i), posD(P_i), SCORname(P_i), Performance(P_i) \rangle$$

$Name(P_i)$  与  $Description(P_i)$  同元素词典中的过程标准化描述相对应;  $In(P_i)$ 、 $Out(P_i)$  描述该过程的输入资源和输出资源及其资源状态;  $Agent(P_i)$  描述该过程的执行部门或执行人,  $Position(P_i)$  描述该过程在元素词典中的位置, 可分解为  $Position = \langle user, parts, genes, specs \rangle$  的形式;  $preD(P_i)$ 、 $posD(P_i)$  描述与该过程相连接的前后依赖标识信息;  $SCORname(P_i)$  是该过程在 SCOR 供应链参考模型中对应的管理过程, 它可以对应计划、资源、制造、分销或退货等标准供应链过程;  $Performance(P_i)$  描述特定供应链过程的绩效, 它根据 SCOR 模型的绩效评价指标体系, 可以分解为  $Performance(P_i) = \{CustomerFacing, InternalFacing\}$  的形式。在这些属性中,  $Name(P_i)$ 、 $Description(P_i)$  和  $Position(P_i)$  使表述的过程与供应链元素词典相对应;  $In(P_i)$ 、 $Out(P_i)$  和  $Agent(P_i)$  描述了该过程资源的使用情况及资源的状态, 保证了过程模型的动态性;  $preD(P_i)$ 、 $posD(P_i)$  是构建供应链模型的基本连接信息;  $SCORname(P_i)$  和  $Performance(P_i)$  使该过程乃至整个供应链流程可以通过 SCOR 模型进行绩效评价和过程优化。

过程元素之间的约束是指过程元素之间能够从一个或几个过程元素地的执行进入下一个或几个过程元素的执行所必须满足的条件。Petri 网中用库所来存储元素之间的变迁所要满足的资源条件, 根据这种原理, 将描述过程元素之间的约束的元素定义为“事件”对象。“事件”的属性可以定义为

$$E_i = \langle Id(E_i), Resource(E_i), preD(E_i), posD(E_i), Description(E_i) \rangle$$

其中  $Resource(E_i)$  描述了前后过程元素之间发生变迁的资源条件, 它是前一个或几个过程元素或是一组资源的集合, 每个资源又包含资源名称和

变迁发生的资源状态条件两个属性. 以上表述可表示为

$Resource(E_i) = \{R_j, j = 1, 2, \dots, R_j \in X, X \text{ is document or material}\}$

$R_j = \langle Name(R_j), OS(R_j) \rangle$

$preD(E_i), postD(E_i)$  描述了与该事件相连接的前后依赖标识信息, 是构建供应链模型的基本连接信息.

供应链过程元素词典使供应链流程中的过程实现了标准化, 但是要进行供应链过程动态建模, 必须在构建元素词典的基础上, 确定供应链过程元素之间的约束和协调关系<sup>[28]</sup>. 协调理论中, 协调被定义为管理活动之间的依赖关系. 通过这种观点, 可以区分不同的依赖关系和不同的管理依赖关系的协同过程. 相同的协同机制可能存在于不同的过程中, 而不同的协同机制也可能被用于管理一个特定的依赖. 所以, 识别依赖和协同机制对于流程再造很重要<sup>[25]</sup>. 为了表现供应链过程元素之间的协调关系, 描述过程元素之间协调关系的元素可以定义为一个名为“依赖”的对象. 把“依赖”的属性可以定义为

$D_i = \langle Id(D_i), In(D_i), Out(D_i), Rule(D_i) \rangle$

其中  $In(D_i)$  和  $Out(D_i)$  描述紧接在该依赖的前面和后面的对象, 它可以是“过程”、“事件”或另一些“依赖”;  $Rule(D_i)$  描述了紧接在该依赖的前面或后面的对象之间的协调关系.  $Rule(D_i)$  可进一步分解为“流”依赖、“与”依赖、“或”依赖和“异或”依赖四种类型. 该属性可表示为集合:

$Rule(D_i) = \{rule | rule \in X, X = \{flow, and, or, xor\}$ . 为了构成供应链流程之间的衔接, 依赖关系必须存在于所有的供应链过程语法元素对象之间, 用以描述过程元素词典的元素对象以及描述过程元素之间的约束的元素对象, 甚至某一些特定的“依赖”都是由“依赖”进行协调的.

### 1.2.2 语法规则

结合供应链过程元素词典、过程语法元素和语法元素的图形化描述符号, 就可以建立基于语法的供应链过程模型. 但是在建模的过程中, 考虑到供应链过程模型的可执行性和各语法元素的不同特性, 模型构建应遵循以下语法规则:

1) 所有的过程 ( $P_i$ ) 和事件 ( $E_i$ ) 有且只有一

个流入 / 流出连接;

2) 每一个过程都必须以事件 ( $E_i$ ) 开始, 并且以事件或过程接口结束;

3) 依赖 ( $D_i$ ) 可以有一个流入连接与多个流出连接, 或者是多个流入连接与一个流出连接;

4) 连接在连接事件与过程时, 作为一种流依赖 ( $D_i$ ) 存在;

5) 单个事件 ( $E_i$ ) 后面不能跟或和异或对象;

6) 流程中的对象符号必须满足“事件 - 依赖 - 过程 - 依赖 - 事件”的顺序.

### 1.3 语法建模方法的巴科斯 - 诺尔范式表述

对于上述这种建模方法可以采用类似巴科斯 - 诺尔范式 (Backus Naur Form) 的格式, 以一个产生式规则集合的形式对供应链流程定义元模型进行定义. 语法的组成包括以下几个部分

- 标志符: < example\_symbol >
- 关键字: EXAMPLE\_KEYWORD
- 产生式标记: ::=
- 特殊字符: [ ] | / \*

采用巴科斯 - 诺尔范式这种标准的格式, 可以将这种语法建模方法表述如下:

#### 1.3.1 过程 (Process)

```

< Process Definition > ::= PROCESS < process id >
[ NAME < name > ]
[ DESCRIPTION < description > ]
[ INPUT < event list > ]
[ OUTPUT < event list > ]
[ SUBFLOW < process id > ] //子流程
[ HIGHLEVEL PROCESS < process id > ] //
父流程
[ AGENT < agent assignment > ] //流程的
执行者
[ < simulation information > ] //流程的资源
消耗
[ PREDEPENDENCY < dependency list > ] //
前相关性
[ POSTDEPENDENCY < dependency list > ]
//后相关性
< agent assignment > ::= < performer expression >
< simulation information > ::= [ < time estimation
> ]
[ COST < cost estimation > ]

```

### 1.3.2 事件 (Event)

所谓事件, 就是通过一个流程显示出来触发某种行为的消息或者请求, 通常可以理解为现实世界中某种状态的改变. 一般有如下情况:

- 1) 能够触发某个流程开始的外部改变 (例如, 客户订单到达);
- 2) 流程内部处理状态的改变 (例如, 产品制造完毕);
- 3) 带有外部影响的最终结果 (例如, 订单送至客户方).

```

< Event Definition > ::= EVENT < event id >
  [ NAME < name > ]
  [ DESCRIPTION < description > ]
  [ PRODUCEDBY < process list > ]
  [ CONSUMEDBY < process list > ]
  [ PREDEPENDENCY < dependency list > ]
  [ POSTDEPENDENCY < dependency list > ]

```

### 1.3.3 依赖 (Dependency)

依赖描述了流程与流程之间的连接关系.

```

< Dependency Definition > ::= DEPENDENCY <
dependency id >
  [ INPUT < event list > | < process list > ]
  [ OUTPUT < event list > | < process list > ]
  [ RULE < rule > ]
< rule > ::= LOGIC < LOGIC characterization >
  [ TYPE < TYPE characterization > ]
< LOGIC characterization > ::= AND | OR | XOR
< TYPE characterization > ::= JOIN | SPLIT

```

## 2 基于语法的供应链流程定义元模型

在上述语法建模方法的定义和规则的基础上, 可以提炼出一个基于语法的供应链流程定义元模型. 供应链流程定义元模型是描述供应链流程的一个标准化的语言格式, 所有遵循它的流程建模工具所建立的模型都可以通过流程定义语言转换成统一的 XML 格式, 从而实现流程的查询、管理以及不同方法描述的供应链流程之间的转换<sup>[29]</sup>.

XML 是一种极具扩展性和可操作性的标记语言<sup>[30]</sup>, 为了提高 SCPDL 的可操作性、扩展性和结构性, 本研究使用 XML 对规格进行重写, 形成了基于 XML 的供应链流程定义语言, 简称 XSCPDL (XML

SCPDL), 它把 XML 作为实现流程查询和不同流程交互的平台.

在从语法建模方法中提炼供应链流程定义元模型的过程中, 做了如下的处理: 首先, 为了防止过程与过程的内部嵌套, 使用了过程和功能来表示内外两个视角的过程, 当过程 A 在一个状态下只是作为子过程时, 用功能来表示; 当打开过程 A, 察看过程 A 里的子过程和活动时, 将其表示为过程. 第二, “依赖”关系在建模方法中共有四类, 即“流”关系, “与”关系, “或”关系和“异或”关系, 将“流”关系处理为控制流, 将“与”、“或”和“异或”关系看作规则来进行处理.

于是, 提出了供应链流程定义元模型 (Supply Chain Process Definition Metamodel), 如图 2 所示. 供应链流程定义元模型描述了供应链流程模型的上层实体和内在联系. 它可以用于描述供应链流程模型内部包含的各个对象、对象之间的关系以及对象的属性. 这个元模型有利于建立可以在多个供应链流程产品之间交换信息的模型.

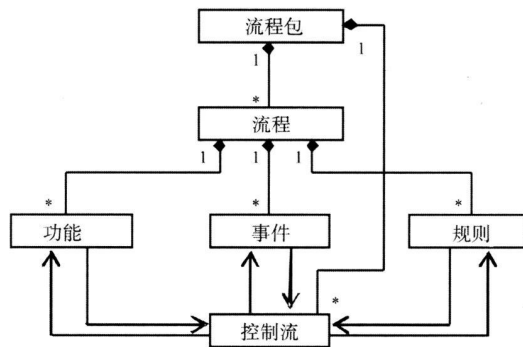


图 2 供应链流程定义元模型

Fig. 2 Supply chain process definition metamodel

供应链流程定义元模型是流程定义语言的重要内容, 它包含了流程、事件、功能、规则和控制流五个基本实体. 下面将逐一进行说明, 并用 XML 将其进行重写.

### 2.1 流程

流程 (process) 定义为供应链中某一增值环节的抽象 (例如, 采购、生产、销售等)<sup>[31]</sup>. 在 XSCPDL 中, 流程实体是一个容器实体, 它包含功能、事件、规则和控制流等其他实体, 并为这些实体提供了流程管理信息和流程执行相关数据. 流程集 (processes) 是流程的集合. 图 3 所示为流程集、流程及其所包含实体的关系图; 表 1 给出了流程表达的属性集.

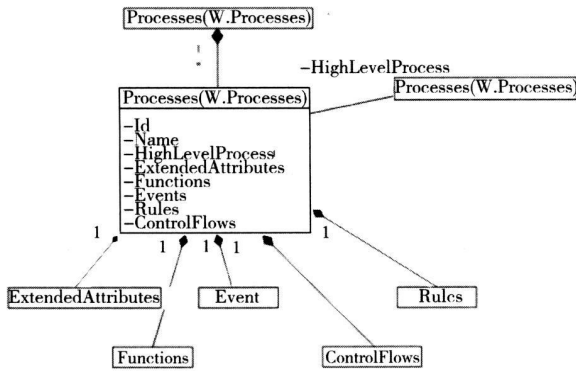


图 3 流程集、流程及其所包含实体的关系

Fig 3 processes process and its related entities

表 1 流程的属性集

Table 1 Attributes of process

	描述
ID	一个供应链流程的标识号, 具有唯一性
Name	供应链流程的名称
High Level Process	指明父流程的 Id 若 Id 为 - 1 则表示是顶层流程.
Access Level	流程的访问级别可以是 PUBLIC 或者 PRIVATE. PUBLIC 可以被外部系统或者应用程序调用; PRIVATE 只能被子流程调用.
Extended Attributes	可选的、用户自定义的属性, 用来满足特殊需求
Events	组成流程的事件列表
Functions	组成流程的功能列表
Rules	组成流程的规则列表
ControlFlows	组成流程的控制流列表

以下是用 XML 对其进行重写, 而形成的 XSCPDL 语言模型:

```
< xsd:element name = "WorkflowProcess" >
< xsd:complexType >
< xsd:sequence >
< xsd:element ref = "ProcessHeader" />
< xsd:element ref = "RedefinableHeader"
minOccurs = "0" />
< xsd:element ref = "FormalParameters"
minOccurs = "0" />
< xsd:element ref = "DataFields" minOccurs = "0" />
< xsd:element ref = "Participants"
minOccurs = "0" />
< xsd:element ref = "Applications"
minOccurs = "0" />
< xsd:element ref = "ActivitySets" />
```

```
minOccurs = "0" />
< xsd:element ref = "Activities" minOccurs =
"0" />
< xsd:element ref = "Transitions"
minOccurs = "0" />
< xsd:element ref = "Events" minOccurs =
"0" />
< xsd:element ref = "Functions" minOccurs =
"0" />
< xsd:element ref = "Rules" minOccurs =
"0" />
< xsd:element ref = "ControlFlows"
minOccurs = "0" />
< xsd:element ref = "ExtendedAttributes"
minOccurs = "0" />
< /xsd:sequence >
< xsd:attribute name = "Id" type = "xsd:
NMTOKEN" use = "required" />
< xsd:attribute name = "Name" type
= "xsd:string" />
< xsd:attribute name = "HighLevelProcess"
type = "xsd:NMTOKEN"
use = "required" />
< xsd:attribute name = "AccessLevel" >
< xsd:simpleType >
< xsd:restriction base = "xsd:NMTOKEN" >
< xsd:enumeration value = "PUBLIC" />
< xsd:enumeration value = "PRIVATE" />
< /xsd:restriction >
< /xsd:simpleType >
< /xsd:attribute >
< /xsd:complexType >
< /xsd:element >
< xsd:element name = "WorkflowProcesses" >
< xsd:complexType >
< xsd:sequence >
< xsd:element ref = "WorkflowProcess"
minOccurs = "0"
maxOccurs = "unbounded" />
< /xsd:sequence >
< /xsd:complexType >
< /xsd:element >
```

### 2.2 事件

事件 (Event) 表达触发某种行为的消息或请求, 或者是某种行为产生的结果. 事件通常可以理解为现实世界中某种状态的改变. 事件集 (Events) 则是事件的集合.

在 XSCPD L 中, 事件描述功能的前提条件或者后果. 所谓前提条件是指在一个功能之间必须出现或已经发生了的事情, 而后果就是一个功能的结果. 事件可以是某人为事件或者是计算机系统操作的结果. 此外每一个流程的最终事件, 除了作为本流程的一个重点外, 还可以作为下一流程的触发事件, 以这种方式就可以将流程的不同部分通过事件连接起来, 形成端到端的流程图. 图 4 给出了事件集、事件及其属性集的关系; 表 2 所示为事件的属性集

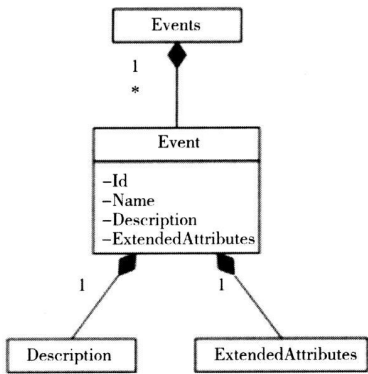


图 4 事件集、事件及其属性集的关系

Fig. 4 Events, event and its attributes

表 2 事件的属性集

Table 2 Attributes of function

	描述
ID	事件的 ID.
Name	事件的名称.
Description	事件的简短文本描述.
Extended Attributes	可选的、用户自定义的属性, 用来满足特殊需求.

也可以用 XML 对其进行重写 (因篇幅所限, 省略, 下同).

### 2.3 功能

功能 (Function) 表示流程中的某个行为或者完成特定任务的活动. 功能集 (Functions) 是功能的集合. 在 XSCPD L 中, 功能是流程中的增值活动. 对应于事件, 功能可以由人或者由计算机系统完成. 每一个功能都包含输入 (信息或者物料), 经过处理后创造输出 (不同的信息或者是产品), 同时在处理过程中可能会消耗一定的资源.

一个功能也许是一个子流程, 在这种情况下, 该活动是一个特定的、分离的流程定义的执行容器. 流程定义允许在子流程中包含自己的功能定义、控制流、规则定义、资源和应用程序分配 (即使这些是从父流程中继承而来). 图 5 表达了功能集、功能及其属性集的关系; 表 3 所示为功能的属性集.

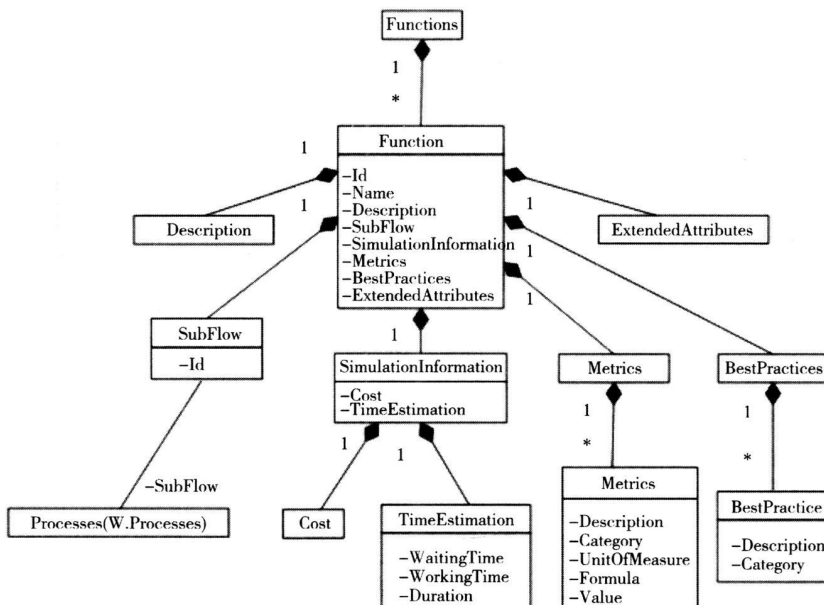


图 5 功能集、功能及其属性集的关系

Fig. 5 Functions, function and its attributes

表 3 功能的属性集  
Table 3 A ttributes of events

	描述
ID	功能的 ID.
Nam e	功能的名称.
Description	功能的简短文本描述.
Ex tended A ttributes	可选的、用户自定义的属性, 用来满足特殊需求.
Sub F low	由其他流程实现功能, 该流程由 ID 指定. 若 ID 为 - 1 则说明没有子流程.
S imulation Information	用于流程仿真的信息, 包括成本和时间估计. 其中时间估计包括等待时间、工作时间、持续时间.
M etrics	衡量功能绩效的绩效指标列表. 这些指标被用来反映一个具体功能的性能水平, 并可以定义其计算方法. 例如, 针对“安排产品配送”功能, 可以使用“平均变更周期”等指标来衡量.
Best Practices	功能的最佳实践的列表. 最佳实践反映产业内优秀的管理方式和解决方案. 例如针对“安排产品配送”功能, “使用 EDI 交易”是一项优秀的解决方案.

### 2 4 规则

规则 (Rule) 用于构建复杂的流程流转条件. 例如:

```

IF condition1 THEN
    TO controlflow1
ELSE IF condition2 THEN
    TO controlflow2
ELSE
    TO controlflow3
ENDIF
    
```

其中的 condition 由控制流 (ControlFlow) 的 Xpression 属性确定. 规则集 (Rules) 是规则的集合. 在 XSCPD L 中, 有三种基本的规则: 与、或、异或.

“与”规则 (AND). 在功能之前表示与分支, 流程被分为两个或多个并行的分支. 在功能之后表示与事件, 所有的事件要同时满足才能触发功能. “或”规则 (OR). 在功能之前表示或决策, 在一个决策之后有一个或多个可能的结果路径. 在功能之后表示或事件, 功能有一个或多个触发事件. “异或”规则 (XOR). 在功能之前表示异或决策, 在某一时刻有且只有一个可能的路径. 在功能

之后表示异或事件, 在某一时刻有且只有一个可能的触发事件.

除此之外, 三个基本规则加以组合还可以反映更复杂的规则. 例如, 或 / 与、或 / 异或、异或 / 与、异或 / 或、与 / 或、与 / 异或. 图 6 所示为规则集、规则及其属性集的关系; 表 4 给出了规则的属性集.

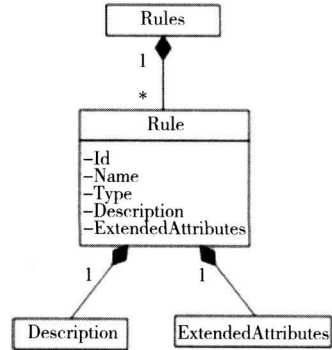


图 6 规则集、规则及其属性集的关系

Fig 6 Rules, rule and its attributes

表 4 规则的属性集

Table 4 A ttributes of rules

	描述
ID	规则的 ID.
Nam e	规则的名称.
Type	规则的类型, 包括 AND、OR、XOR.
Description	规则的简短文本描述.
Ex tended A ttributes	可选的、用户自定义的属性, 用来满足特殊需求.

### 2 5 控制流

控制流 (ControlFlow) 描述了流程的流转路径, 也描述在流程运行期间流转发生与否的条件. 控制流与规则相结合可以构成复杂的控制和构造约束. 控制流集 (ControlFlow) 是控制流的集合.

一个流程的定义, 可以看作是由结点 (包括事件、功能和规则) 和边 (即控制流) 构成的图结构. 边都是有向的, 而且由一对结点来决定 (源结点、目标结点).

控制流可能会被标上流转条件. 标有条件的控制流, 当条件满足时发生流转. 如果没有标记流转条件, 那么就当作流转条件永远为真.

在 XSCPD L 中, 功能和事件永远只有一个输



入和输出控制流, 规则可以有多个输入控制流或者多个输出控制流. 用于表示控制流控制约束的选项由规则的类型决定.

图 7 所示为控制流集、控制流及其属性集的关系; 表 5 表达了控制流的属性集.

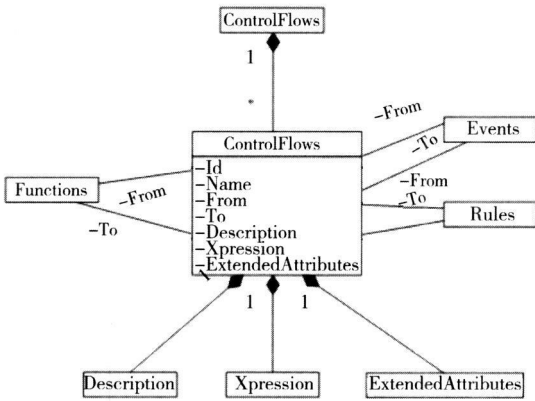


图 7 控制流集、控制流及其属性集的关系

Fig 7 Controlflows, controlflow and its attributes

表 5 控制流的属性集

Table 5 Attributes of controlflow

	描述
ID	控制流的 ID.
Name	控制流的名称.
From	控制流的源结点的 ID. 源结点可以是功能、事件或规则.
To	控制流的目标结点的 ID. 目标结点可以是功能、事件或规则.
Description	控制流的简短文本描述.
Xpression	条件表达式, 描述控制流的流转条件.
Extended Attributes	可选的、用户自定义的属性, 用来满足特殊需求.

### 3 基于语法的供应链流程建模方法的应用范例

在基于语法的建模方法以及供应链流程定义语言的基础上, 本研究设计开发了一种建模方法测试工具 SCW izard. SCW izard 不仅实现了建模功

能, 而且能够将供应链实例流程按照供应链流程定义语言进行转化, 转化后的 XML 将是一种结构化的通用标记语言, 可以用于计算机系统的存储、查询. 可以用一个非常简单的范例来进一步说明并验证 SCPDL 的可行性. 实验流程是制造业供应链过程中的一段, 如图 8 所示.

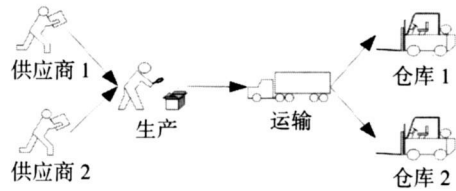


图 8 一个制造业供应链的局部流程

Fig 8 Partial supply chain process

这段供应链流程非常简单且经典. 首先制造商向供应商 1 和供应商 2 购买原材料, 然后使用原材料进行生产活动, 制造出的产品再由物流配送到两个仓库中去. 通过语法建模所建立的模型基本信息可以用表 6 来进行表示. 表 6 中所显示的各属性对于表述动态协调关系以及语法模型的柔性至关重要. 根据实验流程, SCW izard 可以建立相应的语法模型, 并实现语法模型与 XML 标准文档的转化 (如图 9 所示).

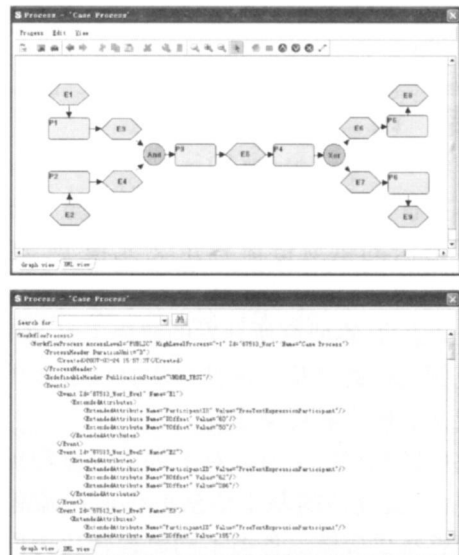


图 9 用 SCW izard 工具构建的供应链范例流程及 XML 接口输出

Fig 9 Supply chain process built by SCW izard and its XML interface

表 6 供应链范例流程的基本信息表  
Table 6 Basic information of supply chain process case

P	Name	Agent	In Out	PreD PostD
P1	生产原料 1	供应商 1	原料 1 订单; 原料 1 及其运单	D1 D2
P2	生产原料 2	供应商 2	原料 2 订单; 原料 2 及其运单	D3 D4
P3	生产	生产部门	原料 1 /原料 2 产品	D5 D6
P4	运输	运输部门	产品; 产品	D7 D8
P5	存储产品到仓库 1	仓库 1	产品; 库存产品	D9 D11
P6	存储产品到仓库 2	仓库 2	产品; 库存产品	D9 D12
E	Description	Resource < Name RS >	PreD PostD	
E1	原料 1 订单已收到	< 原料 1 订单, 供应商 1 收到 >	Null D1	
E2	原料 2 订单已收到	< 原料 2 订单, 供应商 2 收到 >	Null D2	
E3	原料 1 及运单已收到	< 原料 1 运单, 收到 >	D2 D5	
E4	原料 2 及运单已收到	< 原料 2 运单, 收到 >	D4 D5	
...	...	...	...	...
D	In	Out	Rule	
D1	E1	P1	流规则	
D2	P1	E3	流规则	
...	...	...	...	
D5	E3/E4	P3	与规则	
...	...	...	...	
D8	P4	E6 /E7	与或规则	
...	...	...	...	...

## 4 结 论

本文首先在语法建模方法的基础上提出了一个基于语法的供应链流程定义元模型. 然后用 XML 对其规则进行重写, 从而形成了一套供应链流程定义语言. 本研究所开发的建模工具 SCWizard 对于范例流程的实验展现进一步验证了该供应链流程定义元模型的可操作性.

供应链流程定义语言对于供应链流程再造至关重要. 本文设计的基于语法的供应链流程定义语言能够将供应链上各类不同流程转化为同一种

XML 的通用格式, 实现了供应链流程模型与 XML 数据形式之间的相互映射. 这样不同软件提供商提供的建模工具所建立的供应链流程模型即可以在 XML 这个公共平台上进行对话和交流. 用 XML 形式的流程数据表达同样能支持对供应链流程实现便利地搜索、存储和仿真.

进一步研究将着重于两个方面. 一方面是采集足够的供应链流程案例, 形成行业供应链流程案例库, 并进一步完善流程定义语言. 另一方面运用 CBR ( Case-Based Reasoning ) 案例推理方法来实现供应链流程再造的知识库构建和决策支持.

### 参 考 文 献:

[ 1 ] Grover V, Fiedler K D, Teng J T C. Exploring the success of information technology enabled business process reengineering [ J ]. IEEE Transactions on Engineering Management, 1994, 41( 3 ): 276- 284.  
[ 2 ] Patayakuni R, Rai A, Seh N. Relational antecedents of information flow integration for supply chain coordination [ J ]. Journal of Management Information Systems, 2006, 23( 1 ): 13- 49.

- [ 3] Broadbent M, Weill P, Chair D S, Keamey A T. The implications of information technology infrastructure for business process redesign[ J]. *Management Information Systems Quarterly*, 1999, 23( 2): 159- 182
- [ 4] 万 猛, 赵锡斌, 胡飞虎. 业务流程网建模技术及应用研究[ J]. *管理科学学报*, 2007, 10( 6): 41- 48  
Wan M ang, Zhao Xi bin, Hu Fei hu. Research on business process net modeling technology and application[ J]. *Journal of Management Sciences in China*, 2007, 10( 6): 41- 48
- [ 5] Choi I, Jeong H, Song M, Ryu Y U. IBM-EPDL: An XML-based executable process definition language[ J]. *Computers in Industry*, 2005, 56( 1): 85- 104
- [ 6] W MC. Interface: Process Definition Interchange Process Model W MC Standards W MC-TC-1016P, <http://www.wfn.c.org>, 1999
- [ 7] W MC. Interoperability W FXML Binding W MC Standards W MC-TC-1023, <http://www.wfn.c.org>, 2000
- [ 8] W MC. Workflow Process Definition Interface—XML Process Definition Language (XPDL), W MC Standards W MC-TC-1025, <http://www.wfn.c.org>, 2001
- [ 9] Pentland B T. Grammatical models of organizational processes[ J]. *Organization Science*, 1994, 6( 5): 541- 556
- [ 10] Kettinger W J, Teng J T C. Business process change: A study of methodologies, techniques, and tools[ J]. *Management Information Systems Quarterly*, 1997, 21( 1): 55- 98
- [ 11] Weick K E. *The Social Psychology of Organizing*[ M ]. Second Edition, New York: Random House, 1979
- [ 12] Chomsky N. *Knowledge of Language: Its Nature, Origin, and Use*[ M ]. New York: Praeger, 1986
- [ 13] Abbott A. A primer on sequence methods[ J]. *Organization Science*, 1990, 1( 4): 375- 392
- [ 14] Salancik G R, Leblebici H. Variety and Form in Organizing Transactions: A Generative Grammar of Organization[ M ]. Nancy D Tomaso and Samuel B Bacharach (Eds). *Research in the Sociology of Organizations* Greenwich, CT: JAI Press, 1988, 1- 32
- [ 15] Thomas W M, Kevin C, Jintae L, et al. Tools for inventing organizations: Toward a handbook of organizational processes [ C ] // *Proceedings of the 2nd IEEE Workshop on Enabling Technologies Infrastructure for Collaborative Enterprises*, Morgantown, WV, April 20- 22, 1993
- [ 16] Drazin R, Sandelands L E. Autogenesis: A perspective on the process of organizing[ J]. *Organization Science*, 1992, 3( 2): 230- 249
- [ 17] Skvoretz J, Fararo T J. Languages and grammars of action and interaction: A contribution to the formal theory of action[ J]. *Behavioral Science*, 1980, 29( 2): 81- 97
- [ 18] Barley S R. Technology as an occasion for structuring: Evidence from the observation of CT scanners and the social order of radiology departments[ J]. *Administrative Science Quarterly*, 1986, 21( 1): 78- 108
- [ 19] Abell P. *The Syntax of Social Life: The Theory and Method of Comparative Narratives*[ M ]. New York: Charendon Press, 1987
- [ 20] Sandelands L E. Task grammar and attitude[ J]. *Motivation and Emotion*, 1987, 11( 2): 121- 143
- [ 21] Coulter J. *Mind in Action*[ M ]. Cambridge: Polity Press, 1989
- [ 22] White H C. Social grammar for culture: Reply to Steven Brint[ J]. *Sociological Theory*, 1992, 10( 2): 209- 213
- [ 23] Thomas W M, Kevin C, Jintae L, et al. Tools for inventing organizations: Toward a handbook of organizational processes [ J]. *Management Science*, 1999, 45( 3): 425- 443
- [ 24] Pentland B T. Grammatical models of organizational processes[ J]. *Organization Science*, 1995, 6( 5): 541- 556
- [ 25] Kobayashi T, Tanaki M, Konoda N. Business process integration as a solution to the implementation of supply chain management system[ J]. *Information & Management*, 2003, 40( 8): 769- 781
- [ 26] Changchien S W, Shen H. Supply chain reengineering using a core process analysis matrix and object-oriented simulation [ J]. *Information & Management*, 2002, 39( 5): 345- 359
- [ 27] Thomas W M, Kevin C, George A H. *Organizing Business Knowledge: The MIT Process Handbook*[ M ]. Cambridge: MIT Press, 2003

(下转第 63 页)

## Relationship between customer satisfaction and shareholder value

WANG Yi<sup>1</sup>, ZHAO Ping<sup>2</sup>

1. Business School, Central University of Finance and Economics, Beijing 100081, China

2. School of Economics and Management, Tsinghua University, Beijing 100084, China

**Abstract** The relationship between customer satisfaction and corporation performance is one of the most important research focuses of management scholars. With the higher intensity of competing, more and more companies conduct customer satisfaction projects to improve their performance. If they can't measure the effect of these projects accurately, they could not distribute the marketing resource efficiently. In this article, we developed a theoretical framework that specifies how customer satisfaction influenced their financial performance. We collected panel data from the Chinese Customer Satisfaction Index (CCSI) and Chinese Public Corporations, and constructed a basic model and a HLM model to testify the relationship between customer satisfaction and corporation performance. Empirically, we found the positive association between the two variables. We also figured out the significant variation across different industries of the association.

**Key words** customer satisfaction; financial performance; panel data; hierarchical linear model

(上接第 43 页)

[28] 李美霞. 供应链过程分类及其元素词典的建立 [D]. 天津: 南开大学, 2006.

Li Meixia. Research on Classification and Lexicon Building of Supply Chain Process [D]. Tianjin: Nankai University, 2006.

[29] JuharMikka N, Paavo K. A review of XML-based supply chain integration [J]. Production Planning & Control, 2004, 15 (6): 608-621.

[30] Elliott R H. Effective XML: 50 Specific Ways to Improve Your XML [M]. CA: Pearson Education, 2003.

[31] Ekinga D J, Horak T, Lee C, Bruner C. Business process management: Survey and methodology [J]. IEEE Transactions on Engineering Management, 1995, 42(2): 119-128.

## Grammar-based supply chain process definition meta-model

YAN Jian-yuan<sup>1</sup>, LI Kai<sup>2</sup>, ZHANG Lu<sup>1</sup>

1. Business School of Nankai University, Tianjin 300071, China

2. Teda College of Nankai University, Tianjin 300071, China

**Abstract** Grammatical Modeling is a latest modeling language with great potentials, which can be the foundation of supply chain process performance evaluation, optimization and redesign. Supply chain process definition language (SCPDL) is undoubtedly a vital part as a standard language format. It can make the supply chain process be converted into the corresponding grammatical model, so as to implement the inquiry- and management of supply chain processes, and to further provide a foundation to transform the supply chain processes described in different methods. This research proposes a grammar-based supply chain process definition meta-model, and rewrites its rules with XML. Then we can design a kind of XML SCPDL, which will actualize the transformation between the supply chain process engine and the modeling management and optimization tools. In addition, it makes the interaction of different supply chain process models provided by different supply chain software vendors become true.

**Key words** supply chain meta-model; grammatical modeling; process definition language; XML