

基于扩展 Petri 网模型的 BPMN 形式化

李宗花^{1,2} 周晓峰¹ 吴克力² 陈伏兵²

(河海大学计算机与信息学院 南京 211100)¹ (淮阴师范学院计算机科学与技术学院 淮安 223300)²

摘要 BPMN(Business Process Modeling Notation)作为一个在系统开发早期阶段获取业务过程模型的标准,指导系统的设计和开发,其模型的正确性是影响软件开发质量的关键。鉴于 BPMN 模型的形式化可以验证模型的正确性,提出了一种利用扩展 Petri 网模型,应用模型驱动技术实现 BPMN 模型形式化自动执行的方法。该方法通过细化 Petri 网模型中的 Transition 和 Place 元素以及增加 Organization Identifier 和 Group Identifier 容器,使其不但能够描述 BPMN 模型中的动态行为,而且还能描述 BPMN 模型中的动态行为协作和静态组织结构。从元模型结构、语法和图标记方面详细分析了扩展的 Petri 网模型元素,利用模型驱动开发技术设计 BPMN 模型元素至扩展的 Petri 网模型元素的转换规则,并在 Eclipse 平台上使用 ATL 模型转换语言执行映射,实现形式化的自动执行。最后在此基础上应用 Travel Agency 系统演示了模型形式化插件 BPMN2ExtendPetrinets 的执行结果。

关键词 BPMN 模型,模型形式化,Petri 网,模型转换

中图分类号 TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.11.008

BPMN Formalization Based on Extended Petri Nets Model

LI Zong-hua^{1,2} ZHOU Xiao-feng¹ WU Ke-li² CHEN Fu-bing²

(College of Computer and Information Technology Engineering, Hohai University, Nanjing 211100, China)¹

(School of Computer Science and Technology, Huaiyin Normal University, Huaian 223300, China)²

Abstract The business process modeling notation(BPMN), as a standard captures business processes in the early phases of system development, instructs the designing and development. The correctness of the BPMN model is a key to influencing software success. In view that the BPMN formal model can verify the correctness of the model, an extended Petri nets model was proposed to apply model-driven development technology to realize formalization automatic execution of the BPMN model. By refining the Transition and Place elements of the basic Petri nets, and increasing the Organization Identifier and Group Identifier container, the model is not only able to describe the dynamic behaviour of the BPMN model, but also can describe the dynamic behaviours cooperation and the static organization structure of the BPMN model. This paper analyzed the elements of the extended Petri net model from metamodel structure, grammar and notation detailed, and used model-driven development to design the transformation rule from BPMN model elements to extended Petri net model elements. In order to carry out the formalization automatic executing, the executing of the transformation rule uses ATL model transformation language and the ATL transformation code runs on the Eclipse framework. Finally, the Travel Agency system was applied to demonstrate the executing result of the formalization plug (BPMN2 Extend Petrinets).

Keywords BPMN model, Model formalization, Petri nets, Model transformations

业务过程模型能够描述业务系统的服务过程,在大多数研究中被用于描述 CIM(Computation Independent Model)层次模型^[1-3],而业务过程相关的建模语言也在 MDA(Model Driven Architecture)不同抽象层次中提出,如以 BPMN^[4]为代表的 CIM 层次模型,以 BPDM(Business Process Definition Metamodel)^[5]为代表的 PIM(Platform Independent Model)层次模型和以 WS-BPML(WebSphere-Business Process Modeling Language)^[6]为代表的 PSM(Platform Special Model)层次模型。其中业务过程建模标记 BPMN2.0 是被 BPM 工业接受的一个标准并被广泛使用,支持从创建流程轮廓的业务

分析到这些流程的最终实现,直到最终用户的管理监控。同时 BPMN2.0 提供了清晰而精准的执行语义来描述元素的操作,还支持直接映射至 WS-BPEL,实现模型的自动执行^[7-10]。

BPMN 语言允许业务分析员和开发者以图标记的方式描述业务系统的过程信息。但其标记描述^[4]并不包括 BPMN 模型的语义,使得 BPMN 模型的正确性和无二义性难以验证。为了验证 BPMN 模型, Wong 等^[11]利用 Z 语言描述 BPMN 模型的语法结构,利用进程代数 CSP 描述其语义结构,并利用 CSP 语义完整地定义了 BPMN 模型之间的相互关系和模型细化的语义过程。但该研究从纯数学的角度来分析

到稿日期:2016-02-01 返修日期:2016-04-21 本文受国家科技支撑计划(2013BAB05B00,2013BAB06B04),国家自然科学基金(41471425),淮安市科技支撑计划(HAS2015005-1)资助。

李宗花(1981-),女,博士生,讲师,主要研究方向为形式化方法和模型驱动开发, E-mail: leeleaf@163.com;周晓峰(1965-),男,博士,教授,主要研究方向为模型驱动开发和复杂数据管理;吴克力(1963-),男,博士,教授,主要研究方向为数字签名和信息安全;陈伏兵(1964-),男,博士,教授,主要研究方向为数字图像处理和模式识别。

2.1 扩展的 Petri 网模型语法定义

CEN 作为基本的 Petri 网类型,由条件(状态)、事件(变迁)、从条件到事件和从事件到条件的连接(箭头)构成^[24]。句法上条件表示为环,事件表示为矩形。事件可以有一组前提条件和后置条件,使用标符(Marking)来标记条件。与其他 Petri 网相比,CEN 具有简单、容易理解的优点。但由于基本的条件事件网重点刻画条件至事件和事件至条件的变迁,采用基本条件事件网形式化 BPMN 模型会导致模型中组织结构视图的遗漏,从而使得形式化业务过程模型不完整。因此,本文采用扩展 CEN 来形式化业务过程模型。

定义 1 一个基本的条件事件网被定义为由一个 5-元组组成^[25]。

$CEN = \langle Cond\text{-}set, Event\text{-}set, PreCond, PostCond, Marking \rangle$
其中, $Cond\text{-}set = \{cond_1, cond_2, cond_3, \dots, cond_n\}$, 表示一组条件的集合; $Event\text{-}set = \{event_1, event_2, event_3, \dots, event_n\}$, 表示一组事件的集合; $PreCond = event_i \xrightarrow{m} Cond\text{-}set$, 表示前置条件是从事件到条件组的映射; $PostCond = event_i \xrightarrow{m} Cond\text{-}set$, 表示后置条件是从事件到条件组的映射; $Marking = cond_i \xrightarrow{m} Mark$, 表示标记是从标注(Mark)到条件的赋值; $Mark = \{empty | token\}$, 表明标记是空或者标符。

在定义过程中往往将 CEN 网设定为良构的,因此其良构的条件事件网:

$WF\text{-}CEN(c, e, prec, postc, mark) \equiv$

- (1) $dom\ prec = e \wedge$
- (2) $dom\ postc = e \wedge$
- (3) $c = \bigcup \{rng\ prec \cup rng\ postc\} \wedge$
- (4) $\forall es; Event\ es \in e \Rightarrow prec(es) \cup postc(es) \neq \{\} \wedge$
- (5) $dom\ mark = c$

其中,(1)和(2)表示条件的集合与前置和后置条件映射的定义集相同;(3)表示每一个条件都是前置条件或后置条件;(4)表示每一个事件至少有一个前置或后置条件;(5)表示标记(mark)包括所有条件。

在以上基本条件事件网的基础上扩展条件和事件等元素,扩展的条件事件网定义如下。

定义 2 一个扩展的条件事件网被定义为由一个 10-元组组成。

$EPN = \langle IP, OP, ST, BT, IA, OA, M_0, SN, OI, GI \rangle$

其中, $IP = \{ip_1, ip_2, ip_3, \dots, ip_n\}$, 表示有限个内部条件的集合; $OP = \{op_1, op_2, op_3, \dots, op_n\}$, 表示有限个外部条件的集合; $ST = \{st_1, st_2, st_3, \dots, st_n\}$, 表示有限个静态事件的集合; $BT = \{bt_1, bt_2, bt_3, \dots, bt_n\}$, 表示有限个动态事件的集合; $IA \subseteq (IP \times ST) \cup (IP \times BT) \cup (OP \times ST) \cup (OP \times BT)$, 表示一组输入弧; $OA \subseteq (ST \times IP) \cup (ST \times OP) \cup (BT \times IP) \cup (BT \times OP)$, 表示一组输出弧; M_0 表示初始标识; $SN = \langle IP, OP, ST, BT, IA, OA, M_0, SN, OI, GI \rangle_{sub}$, 表示子网; OI_i 表示参与 Petri 网模型的一个外部参与者,该参与者包括一系列条件、内部参与者的集合; GI_i 表示参与 Petri 网模型的一个内部参与者,该参与者包括一系列内部条件执行的顺序。

说明:以上定义的 EPN 模型中,元素 IP 和 OP 是对 CEN 模型的 Cond-set 元素的扩展,即将条件细化为内部条件(IP)和外部条件(OP);元素 ST 和 BT 是对 Event-set 元素的扩展,即将事件细化为静态事件(ST)和动态行为事件(BT);

而 OI 和 GI 则是新增加的元素,表示静态组织结构信息。

定义 3 对于任意 Petri 网模型 $EPN = \langle IP, OP, ST, BT, IA, OA, M_0, SN, OI, GI \rangle$, 有 $st_1 \in ST$ 。

(1)若 $\exists ip_1 \in IP$, 且 ip_1 包含了 M_0 标符,有 $ip_1 \times st_1 \neq \emptyset, st_1 \times ip_1 = \emptyset$, 则称 st_1 为 EPN 模型的开始操作,表示为 Petri 网的开始。

(2)若 $\exists ip_1 \in IP$, 且 ip_1 包含了 M_0 标符,有 $st_1 \times ip_1 \neq \emptyset, ip_1 \times st_1 = \emptyset$, 则称 st_1 为 EPN 模型的终止操作,表示为 Petri 网的结束。

(3)若 st_1 既是开始操作,又是终止操作,则称 st_1 为零操作。

定义 4 对于任意 Petri 网模型 $EPN = \langle IP, OP, ST, BT, IA, OA, M_0, SN, OI, GI \rangle$, $\exists \forall st_1, st_2 \in ST$, 设定 $\rho: st_1 \rightarrow st_2$, 表示为 behavior transition 结点 st_1 和 st_2 之间存在顺序关系,表示 st_1 必须在 st_2 之间实施(st_1 为因, st_2 为果)。

其中, $st_1 \rightarrow st_2$ 表示这两个 behavior transition 结点的一个执行顺序对; ρ 表示连接这两个结点的 place 结点。本文定义两个函数 source 和 target 表示连接 place 结点的前驱结点和后继结点,因此有 $source(\rho) = st_1, target(\rho) = st_2$ 。

定义 5 对于任意 Petri 网模型 $EPN = \langle IP, OP, ST, BT, IA, OA, M_0, SN, OI, GI \rangle$, 有 $st_1, st_2, st_3, \dots, st_n \in ST, bt_1, bt_2, bt_3, \dots, bt_n \in BT, \forall ip_1 \in IP, \forall op_1 \in OP$, 满足以下条件:

(1)应用定义 3, st_1 为 EPN 模型的初始操作,且 $IP(st_1 \times ip_1) = IP(ip_1 \times bt_1)$, 表示静态事件结点 st_1 和动态事件结点 bt_1 是一个执行顺序对,表示为 $st_1 \rightarrow bt_1$, 表示 EPN 模型开始执行顺序序列。

(2)应用定义 3, st_1 为 EPN 模型的终止操作,且 $IP(bt_1 \times ip_1) = IP(ip_1 \times st_1)$, 表示静态事件结点 st_1 和动态事件结点 bt_1 是一个执行顺序对,表示 $bt_1 \rightarrow st_1$, 表示为 EPN 模型结束执行顺序序列。

(3)若 $IP(bt_1 \times ip_1) = IP(ip_1 \times bt_2)$, 或者 $OP(bt_1 \times op_1) = OP(op_1 \times bt_2)$, 则表明动态事件结点 bt_1 和 bt_2 是一个执行顺序对,本文用“ \rightarrow ”表示偏序,表示为 $bt_1 \rightarrow bt_2$, 表明 bt_1 在 bt_2 之前实施,因此(BT, \rightarrow)应无环性、组合性及连通性。而多个执行顺序对加上开始执行序列对和结束执行序列对就可以组成一个执行顺序序列 $P_i = \{st_1 \rightarrow bt_1 \rightarrow bt_2 \rightarrow bt_3 \rightarrow \dots \rightarrow st_n\}$, 其中 BT 表示业务系统中的业务任务集, $bt_1, bt_2, bt_3, \dots, bt_n \in BT$ 。

2.2 扩展的 Petri 网元模型和模型标记

2.1 节对 EPN 模型的形式化语法进行了说明,而 EPN 的图标记表示和元模型结构则如表 1 和图 3 所示。可以看出本文所设计的扩展条件事件网模型包括 inner place, outer place, behaviour transition, silent transition, organization identifier 和 group identifier。从元模型结构上可见, Petri 网模型中的 transition 元素在本文中被分为 silent transition 和 behaviour transition。其中 silent transition 表示静态事件,以实心矩形表示,用于表示业务过程的开始和结束,以及捕获业务控制流和消息流的路径信息。而 behaviour transition 表示动态行为事件,以空心矩形标记,用于表示业务过程模型中的活动元素和中间事件信息。Place 元素被分为 inner place 和 outer place, 其中 inner place 以圆标记表示,用于表示不同对象结点之间的控制流信息,以及表示业务对象元素的执行顺序;而 outer place 以虚线圆标记表示,用于表示业务系统

中不同参与者之间的消息流交互信息,以及代表参与者之间的交互协作。

表1 扩展的CEN标记和特征

模型元素	标记	描述
inner place	○	用于链接两个处于同一组织容器的事件;当一个内部条件被标注有标符时,该内部条件就是Petri网模型的开始点
outer place	⊙	一个外部条件用于链接两个处于不同组织容器的事件
behavior transition	□	一个行为事件显示一个操作事件
silent transition	■	一个静态事件显示Petri网模型的开始/结束,以及决策的制定
arc	→	弧元素链接条件和事件
sub Petri nets	□	子Petri网模型与Petri网模型一样拥有相同的模型元素,但链接子Petri网的条件只能为一个
organization identifier	Ol.name {t ₁ , t ₂ , t ₃ , ...}	一个组织容器OI代表Petri网模型中的一个外部参与者;该外部参与者有可能是一个合伙人实体(如一个公司或一个组织),或者是一个通常的合伙人角色(如一个消费者,或一个生产者),还可能是一个物理上的设备
group identifier	Gl.name {t ₁ , t ₂ , t ₃ }	一个小组容器GI代表一个Petri网模型中的内部参与者;这些参与者可能在一个组织容器属于不同的部门和居于不同的职位

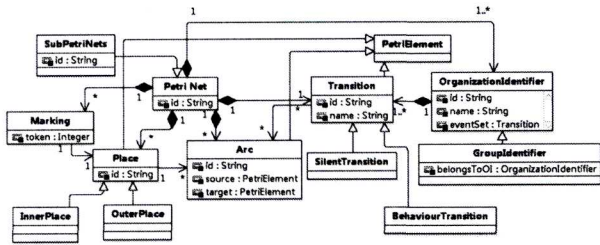


图3 扩展的CEN元模型

值得注意的是,EPN模型中还定义了 *organization identifier* (OI) 和 *group identifier* (GI) 元素代表 Petri 网模型中的外部执行者和内部执行者之间的组织结构特征。其中 OI 代表着不同事件之间的内部条件的容器,规定内部条件不能穿越 OI 的边界。OI 之间的连接依赖于 *outer place*, 因此 *outer place* 可以跨越边界连接相应的事件。

3 BPMN 模型映射至 EPN 模型

3.1 业务过程流说明

BPMN 元素通常可以构造 BPMN 过程图、BPMN 协作图和 BPMN 编排图。这些图表示的模型描述业务系统的内部交互(私有过程)和外部交互(公共过程)。因此,BPMN 模型中设计交互行为的元素包括 *task*, *sub-process*, *call activity*, *event*, *gateway*, *sequence flow* 和 *message flow*。而描述业务系统结构的相关元素如 *pool*, *lane*, *artifact*, *group* 和 *data object* 用于指定合作者实体、或组织和分类活动。

根据 BPMN 说明文档中关于过程流的分析和执行语义分析,业务过程流的主要特征包括以下几点:1)一个开始事件或一个例外事件仅有一个输出流;2)一个结束事件仅有一个输入流;3)一个活动事件或一个中间事件有一个输入流和一个输出流;4)一个分支网关和一个决策网关有一个输入流,多个输出流;5)一个联接网关和一个合并网关有多个输入流和只有一个输出流。

(6)一个子过程只有一个输入流和一个输出流。

3.2 映射规则

根据以上的流特点和 EPN 模型的语法设计,其映射规则如表 2 所列。

表2 BPMN 模型元素映射为 EPN 网模块规则

BPMN 对象 (源模型)	映射规则	EPN 模块 (目标模型)
Start	BPMN 模型中的一个开始事件被映射为 EPN 模型中的带有一个初始内部条件和一个后置内部条件的静态事件。初始内部条件被设置一个标符,且没有输入弧,表示 Petri 网的开始	Ps → SE → Po
End	BPMN 模型中的每一个结束事件被映射为 EPN 模型中带有一个前置内部条件和后置内部条件的静态事件,其中后置内部条件没有输出弧	Pr → EE → Pe
Task1	BPMN 模型中的每一个任务被映射为 EPN 模型中带有一个前置内部条件和一个后置内部条件的动态行为事件	Pr1 → T1 → Po1
Intermediate e	BPMN 模型中的每一个中间事件被映射为 EPN 模型中带有一个前置内部条件和一个后置内部条件的动态行为事件	Pr1 → Ic → Po1
Inclusive	Inclusive 网关中每一个分支被映射为 EPN 模型中带有一个后置内部条件的静态事件	Pr1 → C1 → Po1, Pr1 → C2 → Po2
Merge1	Exclusive 合并网关中每一个输入分支被映射为 EPN 模型中带有一个前置内部条件的静态事件,并且所有的静态事件执行后将合并为一个后置内部条件	Pr1 → M1a → Po1, Pr2 → M1b → Po1
Fork1	Parallel fork 结点被映射为 EPN 模型中带有一个前置内部条件的静态事件,同时每一个输出流分支被映射为一个后置内部条件	Pr1 → F1 → Po1, Po2
Join1	Parallel join 网关中每一个输入流分支被映射为一个前置内部条件,同时 Parallel join 结点被映射为带有一个后置内部条件的静态事件	Pr1 → J1 → Po1, Pr2
Default	Exclusive decision 网关每一条正常路径和缺省路径分别被映射为带有后置内部条件的静态事件	Pr1 → C1 → Po1, Default → Po2
SubProcess1	设定子过程只有唯一的开始事件和结束事件,因此该子过程直接被映射为带前置内部条件的子 EPN 网	Pr1 → SubPage1 → R1
CallActivity1	回调活动中的调用事件和返回事件分别被映射为一个动态行为事件,而回调活动则被映射为一个带有前置内部条件和后置内部条件的子 EPN 网	CA1-call Pr1 → SubPageCA1 → CA1-return Po1
Task1	BPMN 模型中一个带有消息事件的任务,当映射为 EPN 模型时,该任务和消息事件分别被映射为带有后置内部条件的动态行为事件	Pr1 → Task1 → Po1, Em → Po2
Task1, Task2	BPMN 模型中的消息流被映射为 EPN 模型中的外部条件	Pr1 → Task1 → Po1, Ms → Task2 → Po2, Pr2

在 EPN 模型中,一个带有前置条件的静态事件一旦被标记为标符就意味着业务过程的开始,而一个带有前置条件和后置条件的静态事件代表业务序列流执行的路径方向。子过程被映射为一个子 EPN 网模型,代表一个独立的业务过程模型。在模型中,为了避免执行结果的二义性,子过程被严格地限制为只有一个开始事件和一个结束事件。同样,业务过程的回调活动也被看作是一个子过程,通过前置条件的设置开始子过程活动,然后设计一个回调事件用于接收子过程的执行结果,执行结果被定义为子过程的后置条件^[18]。另外,消息流作为 BPMN 模型中两个业务过程之间协作的交互信息,因此将消息流映射为 EPN 模型中的外部条件。其中外部条件的输入弧连接发送消息的动态行为事件,输出弧连接接收消息的动态行为事件。

3.3 元模型执行

由于本文对 Petri 网模型进行了扩展,需要利用 EMF 插件重新定义 Petri 网模型的元模型^[26]。EMF 作为一个元建模框架允许用户定义、编辑和处理一个 ecore 类型的文件(.ecore),图 4 示出了利用 EMF 插件实现的 EPN 元模型定义和 EPN 树状编辑器。

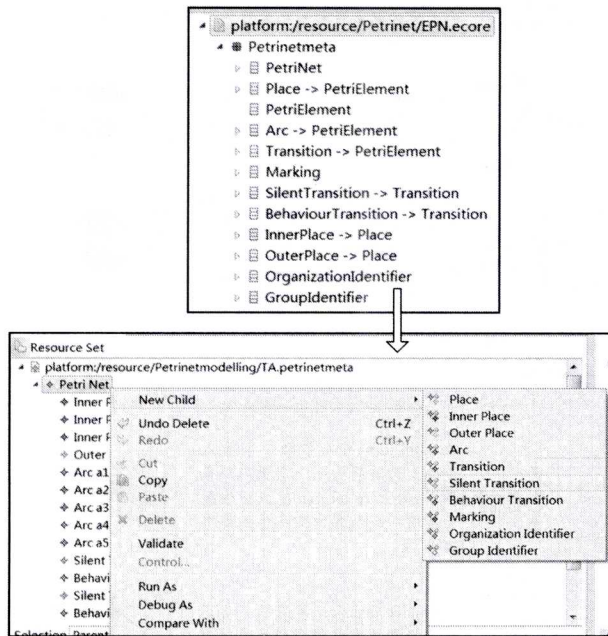


图 4 EPN 元模型定义和 EPN 模型编辑器

3.4 映射执行

本文采用 ATL 代码实现模型形式化的自动执行。ATL 语言由 ATLAS 组织^[27]开发,是一种命令式和申明式相混合且基于 OCL 描述规范、用于表达 MDA 模型转换的语言。从 ATL 语言的执行方法上看,ATL 插件提供了一个 IDE 环境,允许开发者去编辑、调试和编译 ATL 程序^[28]。根据 3.2 节的形式化定义,设计的 BPMN2EPN 模型形式化执行过程如图 5 所示。其详细过程为:首先利用 Activiti BPMN 2.0 插件建立业务系统的 BPMN 模型,然后根据 BPMN 元模型格式将该模型转换为 XMI 类型的文件,该 XMI 文件然后被 BPMN2EPN 转换工具读取。

BPMN2EPN 转换工具的执行过程包括以下 4 个步骤:1)XMI 文件被解析器(Parser)分解为单个独立的 BPMN 对象;2)根据表 2 制定的映射规则,设计 ATL 语言格式的转换

程序;3)ATL 转换程序将单个独立的 BPMN 对象转换为扩展的条件事件网(EPN)模块文件;4)利用组合器将这些转换后的单个独立的 EPN 文件组合为一个完整的 EPN 文件。PNML 文件作为一个标准的采用 XML 交换格式描述 Petri 网的语言,其 ePNK(2012 版本)插件能够在 Eclipse 建模平台上读取 PNML 文件。因此本文利用 PNML 语言描述 EPN 模型,利用 ePNK 插件来处理 EPN 模型。

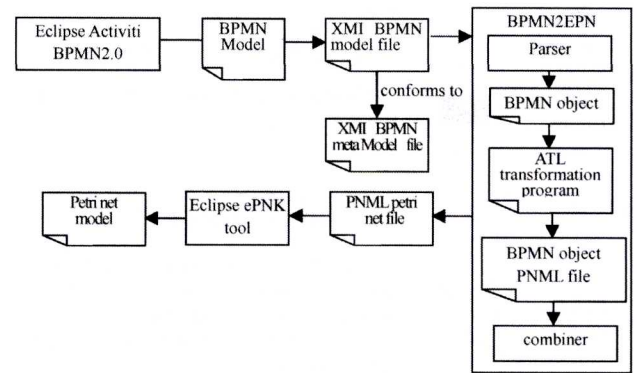


图 5 BPMN 模型形式化为 EPN 模型的过程

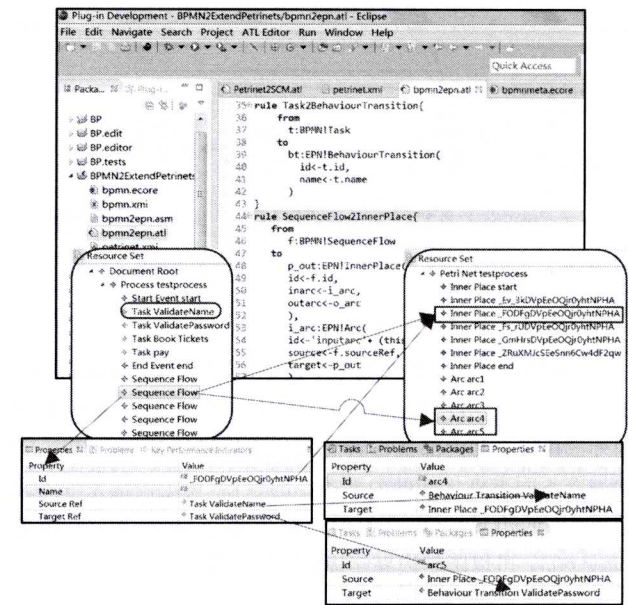


图 6 BPMN 模型至 EPN 模型映射执行部分视图和 BPMN2EPN ATL 代码摘录

为了演示该模型转换的自动执行过程,图 6 显示了 BPMN 模型至 EPN 模型的部分映射执行细节。可以看出,BPMN2ExtendPetrinets 转换插件中定义了 BPMN 模型中的 task 元素至 EPN 模型中的 behaviour transition 元素,以及 sequence flow 元素至 inner place 元素的转换(35 行和 44 行)。在 Task2BehaviourTransition 转换操作中,task 元素的 id 和 name 值被直接转换为 behaviour transition 元素中的 id 和 name 属性值(40-41 行);而与 task 元素相关的 sequence flow 和 message flow 需要转换为 EPN 模型中的 inner place 和 outer place 元素。在 SequenceFlow2InnerPlace 转换操作中,sequence flow 元素的 id 值被直接转换为 inner place 元素的 id 值(49 行)。与此同时 EPN 模型中还创建 in 弧和 out 弧(50-51 行),在创建 in 弧和 out 弧的过程中,需要设计弧连接的源结点和目标结点,如 in 弧的 sourceRef(源结点)为该

sequence flow 的源结点(55 行),而 in 弧的 targetRef(目标结点)为创建的 inner place(56 行)结点。

4 实例演示

本文使用的运行实例来自于 W3C 标准文档^[29] 中的 Travel Agency 业务系统,该业务系统的场景描述如下。

首先,出行者(Customer)向 Travel Agency 提交自己的旅行计划,该计划中包括旅行者打算出行的目的地、出行时间和返回时间、住宿酒店的标准等信息。然后,Travel Agency 接收 Customer 的旅行计划并评估该计划,找到能够为该 Customer 提供服务的 Broker Agent 并连接该 Broker Agent。每一个 Broker Agent 有可能是一个为多个酒店工作的住宿

(Accommodation)代理,也可能是一个为多个交通运输公司工作的交通(Transportation)代理,还可能是为多个旅游景点公司工作的旅游景点(Tourist Attraction)代理。这些公司通过 Broker Agent 提供的服务请求提供相应的服务。然后 Broker Agent 将满足条件的服务传回 Travel Agency, Travel Agency 将这些服务整理后为 Customer 提供一系列满足要求的旅游选项。一旦 Customer 选择了选项中的某一项,他必须提供其信用卡信息给 Travel Agency,这个过程可以称为支付过程,信用卡的支付涉及到相关的金融公司(Finance Company),因此当 Finance Company 验证这个支付是正确有效的以后,Travel Agency 同时要求 Broker Agent 确认订单并发消息通知 Customer。

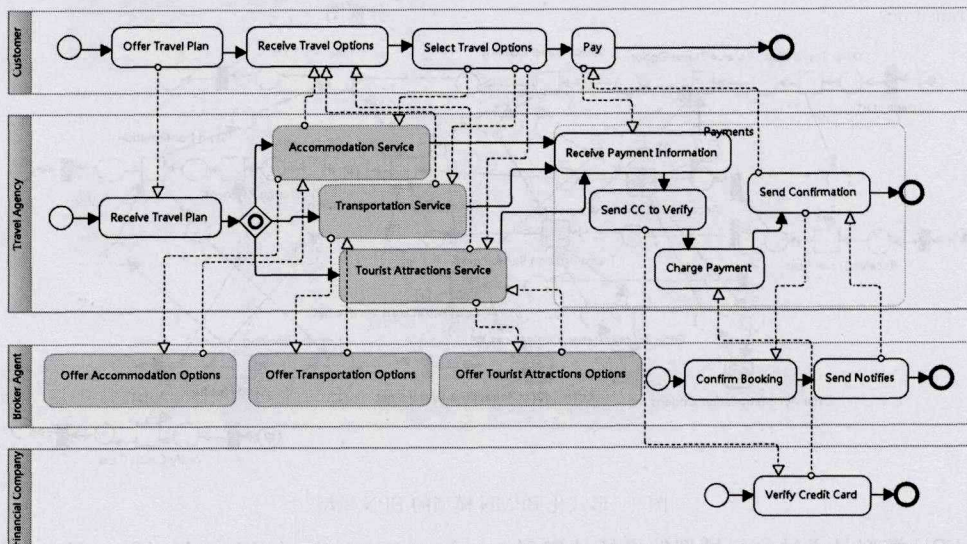


图 7 Travel Agency 系统的完整 BPMN 模型

其完整的 BPMN 模型如图 7 所示。可以看出 Travel Agency 系统中包括 customer, travel agency, broker agent 和 financial company 4 个直接参与者。显然 travel agency 提供了 Accommodation 服务、Transportation 服务和 Tourist Attractions 服务,这些业务以子过程存在于 BPMN 模型中,同时被映射为 EPN 模型中的子网模型。根据模型形式化过程,EPN 模型由 PNML 语言描述。因此模型转换插件 BPMN2ExtendPetrinets 读取的源文件为 xml 格式的 BPMN 模型,执行转换后的目标文件也为 xml 格式的 EPN 模型。图 8 显示了 BPMN 模型形式化为 EPN 模型的 xml 文件内容。

可以看出图 8 中上方 BPMN. xml 文件中第 89 行代码是连接任务 Offer Travel Plan 与 Receive Travel Options 的 sequenceFlow 元素被转换为下方 PNML. xml 文件中的第 19 行的 innerPlace 元素,与此同时创建了 20、21 行的输入输出弧。转换后的 PNML 文件利用 ePNK 读取和编辑该文件,形成图 9 所示的图形格式的形式化 BPMN 模型。

```
<?xml version="1.0" encoding="ASCII"?>
<!-xtools2_universal_type_manager-
<?com.ibm.xtools.emf.core.signature (signature id="com.ibm.xtools.bpmn2.model" version="1.80.4") (feature description="name="com.ibm.xtools.bpmn2.feature" url="" version="1.80.4")/></signature?>
<bpmn2:process id="_RwBDIC-KEeWhbMLL1NrGOA" name="Process1">
```

```
<bpmn2:startEvent id="_mHZL8C-KEeWhbMLL1NrGOA"/>
<bpmn2:task id="_TQdswC-LEeWhbMLL1NrGOA" name="Offer Travel Plan"/>
<bpmn2:task id="_aueWcC-LEeWhbMLL1NrGOA" name="Receive Travel Options"/>
<bpmn2:sequenceFlow id="_vxcac-LEeWhbMLL1NrGOA" sourceRef="_mHZL8C-KEeWhbMLL1NrGOA" targetRef="_TQdswC-LEeWhbMLL1NrGOA"/>
<bpmn2:sequenceFlow id="_nLsPoC-MEeWhbMLL1NrGOA" sourceRef="_aueWcC-LEeWhbMLL1NrGOA" targetRef="_gQNEMC-MEeWhbMLL1NrGOA"/>
<bpmn2:task id="_gQNEMC-MEeWhbMLL1NrGOA" name="Select Travel Options"/>
<bpmn2:task id="_k7rU0C-MEeWhbMLL1NrGOA" name="Pay"/>
<bpmn2:sequenceFlow id="_nvZZoC-MEeWhbMLL1NrGOA" sourceRef="_gQNEMC-MEeWhbMLL1NrGOA" targetRef="_k7rU0C-MEeWhbMLL1NrGOA"/>
</bpmn2:process>
```

(a)

```
<?xml version="1.0" encoding="UTF-8"?>
<pnml xmlns="http://www.pnml.org/version-2009/grammar/pnml">
<net type="http://epnk.tools.pnml.org/signalnet/2012">
<page id="_RwBDIC-KEeWhbMLL1NrGOA">
```

```

<name>
  <text>process1</text>
</name>
<silenttransition id = “_mHXL8C-KEeWhbMLL1NrGOA”/>
<behaviourtransition id = “_TQdswC-LEeWhbMLL1NrGOA”>
  <name>
    <text>Offer Travel Plan</text>
  </name>
</behaviourtransition>
<behaviourtransition id = “_aueWcC-LEeWhbMLL1NrGOA”>
  <name>
    <text>Receive Travel Options</text>
  </name>
</behaviourtransition>

```

```

<innerplace id = “_lx1YC-LEeWhbMLL1NrGOA”/>
<arc id = “inputarc2” source = “_TQdswC-LEeWhbMLL1NrGOA”target = “_lx1YC-LEeWhbMLL1NrGOA”/>
<arc id = “outputarc3” source = “_lx1YC-LEeWhbMLL1NrGOA” target = “_aueWcC-LEeWhbMLL1NrGOA”/>
<behaviourtransition id = “_gQNEMC-MEeWhbMLL1NrGOA”>
  .....
</page>
</net>
</pnml>

```

图8 Travel Agency系统的BPMN模型映射为PNML模型的部分视图

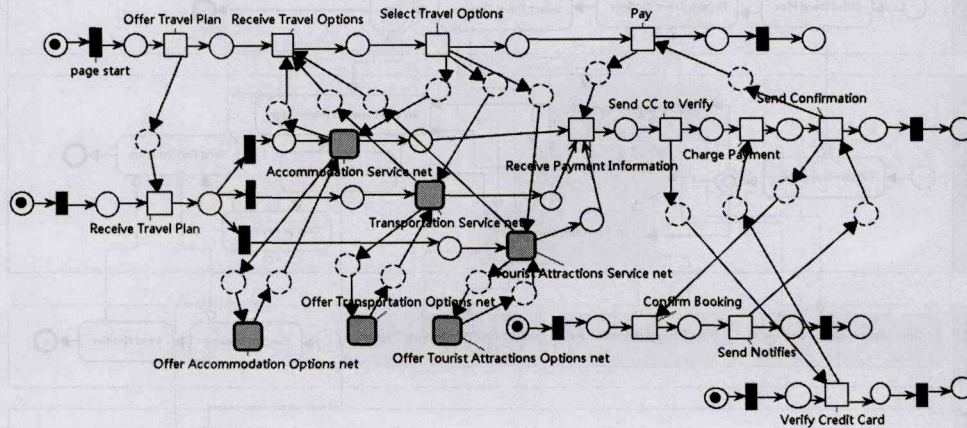


图9 形式化BPMN模型的EPN模型

图9所示的EPN模型是通过执行模型转换插件实现BPMN模型的形式化自动执行结果。第一步通过ePNK工具验证该形式化模型是正确的,不存在死锁、闭环等错误状态;第二步应用2.1节中定义的EPN语法和语义来验证模型的正确性。其语义验证的过程为:首先根据定义3确定图9的开始结点和结束结点,然后根据定义4分析出EPN模型的执行顺序对,最后根据定义5得出如下活动执行顺序序列。

$P1 = \text{page start} \rightarrow \text{Offer Travel Plan} \rightarrow \text{Receive Travel Options} \rightarrow \text{Select Travel Options} \rightarrow \text{Pay} \rightarrow \text{page end}$

$P2 = \text{page start} \rightarrow \text{Receive Travel Plan} \rightarrow \text{Accommodation Service net} \rightarrow \text{Receive Payment Information} \rightarrow \text{Send CC to Verify} \rightarrow \text{Charge Payment} \rightarrow \text{Send Confirmation} \rightarrow \text{page end}$

$P3 = \text{page start} \rightarrow \text{Receive Travel Plan} \rightarrow \text{Transportation Service net} \rightarrow \text{Receive Payment Information} \rightarrow \text{Send CC to Verify} \rightarrow \text{Charge Payment} \rightarrow \text{Send Confirmation} \rightarrow \text{page end}$

$P4 = \text{page start} \rightarrow \text{Receive Travel Plan} \rightarrow \text{Tourist Attractions Service net} \rightarrow \text{Receive Payment Information} \rightarrow \text{Send CC to Verify} \rightarrow \text{Charge Payment} \rightarrow \text{Send Confirmation} \rightarrow \text{page end}$

$P5 = \text{page start} \rightarrow \text{Confirm Booking} \rightarrow \text{Send Notifies} \rightarrow \text{page end}$

$P6 = \text{page start} \rightarrow \text{Verify Credit Card} \rightarrow \text{page end}$

$P7 = \text{page start} \rightarrow \text{Offer Travel Plan} \rightarrow \text{Receive Travel Plan} \rightarrow \text{Accommodation Service net} \rightarrow \text{Offer Accommodation Options net} \rightarrow \text{Accommodation Service net} \rightarrow \text{Receive Travel Options} \rightarrow \text{Select Travel Options} \rightarrow \text{Pay} \rightarrow \text{Receive Payment In-}$

$\text{formation} \rightarrow \text{Send CC to Verify} \rightarrow \text{Verify Credit Card} \rightarrow \text{Charge Payment} \rightarrow \text{Send Confirmation} \rightarrow \text{Confirm Booking} \rightarrow \text{Send Notifies} \rightarrow \text{Send Confirmation} \rightarrow \text{Pay} \rightarrow \text{page end}$

$P8 = \text{page start} \rightarrow \text{Offer Travel Plan} \rightarrow \text{Receive Travel Plan} \rightarrow \text{Transportation Service net} \rightarrow \text{Offer Transportation Options net} \rightarrow \text{Transportation Service net} \rightarrow \text{Receive Travel Options} \rightarrow \text{Select Travel Options} \rightarrow \text{Pay} \rightarrow \text{Receive Payment Information} \rightarrow \text{Send CC to Verify} \rightarrow \text{Verify Credit Card} \rightarrow \text{Charge Payment} \rightarrow \text{Send Confirmation} \rightarrow \text{Confirm Booking} \rightarrow \text{Send Notifies} \rightarrow \text{Send Confirmation} \rightarrow \text{Pay} \rightarrow \text{page end}$

$P9 = \text{page start} \rightarrow \text{Offer Travel Plan} \rightarrow \text{Receive Travel Plan} \rightarrow \text{Tourist Attractions Service net} \rightarrow \text{Offer Tourist Attractions Options net} \rightarrow \text{Tourist Attractions Service net} \rightarrow \text{Receive Travel Options} \rightarrow \text{Select Travel Options} \rightarrow \text{Pay} \rightarrow \text{Receive Payment Information} \rightarrow \text{Send CC to Verify} \rightarrow \text{Verify Credit Card} \rightarrow \text{Charge Payment} \rightarrow \text{Send Confirmation} \rightarrow \text{Confirm Booking} \rightarrow \text{Send Notifies} \rightarrow \text{Send Confirmation} \rightarrow \text{Pay} \rightarrow \text{page end}$

可以看出P1,P2,P3,P4,P5,P6的执行顺序序列都是由inner place条件连接而成,表示业务过程模型中的内部业务流程;而P7,P8,P9这3条执行顺序序列包括了inner place条件和outer place条件,表明这3条序列是外部业务协作过程,即由多个参与者协作实现Accommodation服务、Transportation服务和Tourist Attractions服务。这9条执行顺序序列使得所有的silent transition,behaviour transition,inner place和out place结点都被遍历,且涵盖了Travel A-

gency 系统的所有内部业务过程和外部业务协作过程。

通过以上的 ePNK 语法检查和从语义上的执行顺序序列分析,表明该 BPMN 模型描述的业务过程模型是正确的。当然,BPMN 形式化模型也会出现 ePNK 语法检查不通过的情形(如没有开始事件或结束事件等),或语义检查不通过的情形(如存在闭环或死锁的问题),这两种情形表明该 BPMN 模型是不完整或不正确的。由此可见,本文设计的扩展 Petri 网模型能够实现 BPMN 模型的动态行为和静态组织结构的形式化,确保了系统业务模型的正确性,从而更好地指导软件的设计与开发。

结束语 本文针对现有 BPMN 形式化方法不能解决业务模型的动态行为之间交互协作和静态组织结构的形式化问题,提出了通过扩展 Petri 网模型中的 *Transition* 和 *Place* 元素、增加 *OI* 和 *GI* 元素实现 BPMN 模型的形式化。通过制定 BPMN 模型至 EPN 模型的映射规则,利用 ATL 语言和 EMF 插件在 Eclipse 平台上实现了 BPMN 模型的形式化自动执行。实例演示表明,扩展 Petri 网中的 *Transition* 元素能清晰地描述 BPMN 模型中的操作和决策分支;扩展 *Place* 元素能有效地区分 BPMN 模型的控制流和消息流,从而完整地描述动态行为执行顺序和协作交互;而增加的 *OI* 与 *GI* 元素能完整将 BPMN 模型的组织结构特征形式化。因此,该形式化方法能更加完整地形式化 BPMN 模型,且能通过已开发的形式化插件实现自动执行,使得软件开发人员无需掌握生涩难懂的数学理论即可实现 BPMN 模型正确性的验证,具有确定性的优势。

但为了保障模型转换的正确性以及实现形式化自动执行,文中设置了严格的限制条件,如子过程被严格限制为只有一个开始事件和一个结束事件等。因此,为了能将该方法应用在复杂的业务过程建模中,如何降低限制性约束和实现 BPMN 模型语义自动检测是下一步研究工作需要解决的主要问题。

参考文献

- [1] Touzi J, Benaben F, Pingaud H, et al. A model-driven approach for collaborative service-oriented architecture design [J]. International Journal of Production Economics, 2009, 121(1): 5-20
- [2] Bousetta B, Beggar O, Gadi T. A methodology for CIM modelling and its transformation to PIM [J]. Journal of Information Engineering and Applications, 2013, 3(2): 1-21
- [3] Fabra J, De Castro V, Álvarez P, et al. Automatic execute of business process models: exploiting the benefits of model-driven engineering approaches [J]. Journal of Systems and Software, 2012, 85(3): 607-625
- [4] OMG. Business Process Model and Notation (BPMN) (Version 2.0) [EB/OL]. [2011-01-03]. <http://www.omg.org/spec/BPMN/2.0>
- [5] OMG. Business Process Definition Metamodel (Version 1.0) [EB/OL]. [2008-11-03]. <http://www.omg.org/spec/BPDM/1.0>
- [6] IBM. Business Process modelling Language (BPML) (Version 5.2) [EB/OL]. ftp://public.dhe.ibm.com/software/commerce/doc/sb2bi/v5r2/SI52_BPML_book.pdf
- [7] Harmon P. The OMG's model driven architecture and BPM [J]. Business Process Trends, 2004, 2(5): 1-11
- [8] Feng Xiao-ning, Li Qi-xing, Wang Zhuo. A Mapping Method from BPMN Models to BPEL [J]. Journal of Computer Research and Development, 2013, 50(S1): 44-52 (in Chinese)
- [9] 冯晓宁, 李麒星, 王卓. 一种基于 BPMN 的业务流程图到 BPEL 的映射方法 [J]. 计算机研究与发展, 2013, 50(S1): 44-52
- [9] Wei Ling, Ai Yong-lin, Wei Jun. Model Transformation from BPMN to BPEL2.0 [J]. Application Research of Computers, 2008, 25(11): 3363-3366 (in Chinese)
- [9] 魏凌, 爱永霖, 魏竣. BPMN 到 BPEL2.0 的模型转换方法 [J]. 计算机应用研究, 2008, 25(11): 3363-3366
- [10] Ma Jian, Xu Tao, Zhang Yu-ping. Conversion of business process models based on continuous semantics [J]. Journal of Computer Applications, 2013, 33(S1): 243-246 (in Chinese)
- [10] 马健, 徐涛, 张育平. 基于连续语义的业务流程模型的转换 [J]. 计算机应用, 2013, 33(S1): 243-246
- [11] Wong P Y H, Gibbons J. Formalisations and applications of BPMN [J]. Science of Computer Programming, 2011, 76(8): 633-650
- [12] Dijkman R M, Dumas M, Ouyang C. Semantics and analysis of business process models in BPMN [J]. Information and Software Technology, 2008, 50(12): 1281-1294
- [13] Yang N H, Yu H Q, Sun H, et al. Modeling UML sequence diagrams using extended Petri nets [J]. Telecommunication Systems, 2012, 51(2): 147-158
- [14] Fabra J, Álvarez P, Banares J, et al. DENEBA: a platform for the development and execution of interoperable dynamic web processes [J]. Concurrency and Computation: Practice and Experience, 2011, 23(8): 2421-2451
- [15] Philippi S. Automatic code generation from high-level Petri-Nets for model driven systems engineering [J]. Journal of Systems and Software, 2006, 79(10): 1444-1455
- [16] van der Aalst W M P, ter Hofstede A H M. YAWL: yet another workflow language [J]. Information Systems, 2005, 30(4): 245-275
- [17] Bigham J, Jin X, Gamez D, et al. Hybrid Workflow and Bayesian Networks to Correlate Information in the Protection of Large Scale Critical Infrastructures [J]. Electronic Notes in Theoretical Computer Science, 2005, 121: 87-99
- [18] Li Z H, Zhou X F, Gu A H, et al. A complete approach for CIM modelling and model formalizing [J]. Information and Software Technology, 2015, 65(9): 39-55
- [19] Li Zong-hua, Zhou Xiao-feng, Gu Ai-hua, et al. Survey of CIM modeling and model formalization [J]. Application Research of Computers, 2014, 31(10): 2896-2901 (in Chinese)
- [19] 李宗花, 周晓峰, 顾爱华, 等. CIM 建模及模型形式化方法研究综述 [J]. 计算机应用研究, 2014, 31(10): 2896-2901
- [20] De Castro V, Marcos E, Vara J M. Applying CIM-to-PIM model transformations for the service-oriented development of information systems [J]. Information and Software Technology, 2011, 53(1): 87-105
- [21] Kostin A E. Reachability analysis in T-invariant-less Petri Nets [J]. IEEE Transactions on Automatic Control, 2003, 48(6): 1019-1024
- [22] Yoo T, Jeong B, Cho H. A Petri nets based functional validation for services composition [J]. Expert Systems with Applications, 2010, 37(5): 3768-3776

- [23] Balaban M, Maraee A, Sturm A, et al. A pattern-based approach for improving model quality [J]. *Software System Model*, 2015, 14(4):1527-1555
- [24] Ye Y, Jiang Z B, Diao X D, et al. Extended event-condition-action rules and fuzzy Petri nets based exception handling for workflow management [J]. *Expert Systems with Applications*, 2011, 38(9):10847-10861
- [25] Dines Bjørner. *Software Engineering 2 Specification of Systems and Language*[M]. Springer, 2006:316-322
- [26] Steinberg D, Budinsky F, Paternostro M, et al. EMF: Eclipse Modeling Framework(2nd Edition) [M]. Addison-Wesley Professional, 2008:104-124
- [27] ATLAS. ATL recognized as a standard solution for model transformation in Eclipse [EB/OL]. [2007-01-15]. <http://www.eclipse.org/gmt>
- [28] Jouault F, Allilaire F, Bezivin J, et al. ATL: a model transformation tool [J]. *Science of Computer Programming*, 2008, 72(1/2):31-39
- [29] W3C. Web Service Choreography Interface(WSCI)(Version1.0) [EB/OL]. www.w3.org/TR/wsci
-
- (上接第 11 页)
- [35] Xiong An-ping, Huang Rong, Zou Yang. A kind of HDFS small files storage strategy based on hybrid index [J]. *Journal of Chongqing University of Posts and Telecommunications(Natural Science Edition)*, 2014, 27(1):97-102(in Chinese)
熊安萍, 黄蓉, 邹样. 一种基于混合索引的 HDFS 小文件存储策略[J]. *重庆邮电大学学报(自然科学版)*, 2014, 27(1):97-102
- [36] Zhu Yuan-yuan, Wang Xiao-jing. HDFS optimization program based on GE coding [J]. *Journal of Computer Applications*, 2013, 33(3):730-733(in Chinese)
朱媛媛, 王晓京. 基于 GE 码的 HDFS 优化方案[J]. *计算机应用*, 2013, 33(3):730-733
- [37] Song Bao-yan, Wang Jun-Lu, Wang Yan. Optimized Storage Strategy Research of HDFS Based on Vandermonde Code[J]. *Chinese Journal of Computers*, 2015, 9(38):1826-1836(in Chinese)
宋宝燕, 王俊陆, 王妍. 基于范德蒙码的 HDFS 优化存储策略研究[J]. *计算机学报*, 2015, 9(38):1826-1836
- [38] Lu Mei-lian, Zhu Liang-liang. Load Balancing Strategy Based on CMM Model in HDFS[J]. *Journal of Beijing University of Posts and Telecommunications*, 2014, 10(37):20-25(in Chinese)
卢美莲, 朱亮亮. 基于 CMM 模型的 HDFS 负载均衡策略[J]. *北京邮电大学学报*, 2014, 10(37):20-25
- [39] Xi Ping, Xue Feng. Replica Placement Strategy Based on Multi-layer Consistent Hashing in HDFS [J]. *Computer Systems & Applications*, 2015, 24(2):127-133(in Chinese)
席屏, 薛峰. 多层一致性哈希的 HDFS 副本放置策略[J]. *计算机系统应用*, 2015, 24(2):127-133
- [40] Kun L, Dai Dong, Sun Ming-ming. HDFS+: Concurrent Writes Improvements for HDFS[C]//Proc of IEEE International Conference on Big Data. 2013:182-183
- [41] Islam N S, Lu X, Wasi-ur-Rahman M, et al. In-Memory I/O and Replication for HDFS with Memcached: Early Experiences[C]//Proc of IEEE International Conference on Big Data. 2014:213-218
- [42] Ekanayake J, Li H, Zhang B, et al. Twister: A runtime for iterative mapreduce [C]//Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing. ACM, 2010:810-818
- [43] Abouzeid A, Bajda-Pawlikowski K, Abadi D J, et al. HadoopDB: An architectural hybrid of MapReduce and DBMS technologies for analytical workloads [J]. *PVLDB*, 2009, 2(1):922-933
- [44] Abouzeid A, Bajda-Pawlikowski K, Adadi D J, et al. HadoopDB in action: Building real world applications[C]//Elmagarmid AK, Agrawal D, eds. Proc of th SIGMOD. Indiana: ACM Press, 2010:1111-1114
- [45] Qin X, Wang H, Li F, et al. Beyond Simple Integration of RD-BMS and MapReduce—Paving the Way toward a Unified System for Big Data Analytics: Vision and Progress[C]//2012 Second International Conference on Cloud and Green Computing (CGC). IEEE, 2012:716-725
- [46] Talbot J, Yoo R M, Kozyrakis C. Phoenix: modular MapReduce for shared-memory systems[C]//Proceedings of the Second International Workshop on MapReduce and its Applications. ACM, 2011:9-16
- [47] Xu T, Wang D S, Liu G D, Banian. A Cross-Platform Interactive Query System for Structure Big Data[J]. *Tsinghua Science and Technology*, 2015, 7(11):62-71
- [48] Anderson Q. *Storm real-time processing cookbook* [M]. Birmingham: Packt Publishing, 2015
- [49] Jin Yong-chao, Wu Huai-gu. Research on the Big Data Process Framework Based on Storm and Hadoop[J]. *Modern Computer*. 2015(3):1419-1423(in Chinese)
靳永超, 吴怀谷. 基于 Storm 和 Hadoop 的大数据处理架构的研究[J]. *现代计算机*, 2015(3):1419-1423
- [50] Lee R, Luo T, Huai Y, et al. Ysmart: Yet Another Sql-to-mapreduce Translator[C]//2011 International Conference on Distributed Computing Systems(ICDCS). IEEE, 2011:25-36
- [51] Zhao Huan, Chen Xi. Chinese Tourism Information Search Platform based on Cloud Computing [C]//International Industrial Informatics and Computer Engineering Conference (IIICEC 2015). Beijing, 2015:1236-1240
- [52] Chen Hong. Research on Chinese segmentation algorithm based on Hadoop cloud platform[C]//Information Technology and Mechatronics Engineering Conference(ITOEC 2015). 2015:134-138
- [53] Hadoop 各商业发行版之比较[OL]. [2016-1-10]. <http://www.cnblogs.com/iceTing/p/3392362.html>
- [54] 翟周伟. *Hadoop 核心技术*[M]. 北京:机械工业出版社, 2015:4
- [55] Hadoop 目前在国内外的现状介绍[OL]. [2015-9-10]. http://www.thebigdata.cn/Hadoop/14142.html?utm_source=tuicool
- [56] FusionInsight 大数据平台[OL]. [2016-1-10]. <http://carrier.huawei.com/cn/products/IT/cloud-computing/fusioninsight>