

6-4-2020

Business Process Modeling Algorithm Based on Ontology Language

Huang Ying

1. State Key Laboratory of Software Engineering, School of Computer, Wuhan University, Wuhan 430072, China;;2. Institute of Mathematical and Computer Sciences, Gannan Normal University, Ganzhou 341000, China;;

Kangshun Li

3. College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China;

Li Wei

3. College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China;

Keqing He

1. State Key Laboratory of Software Engineering, School of Computer, Wuhan University, Wuhan 430072, China;;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Business Process Modeling Algorithm Based on Ontology Language

Abstract

Abstract: In the view of problem in lack of information sharing mechanisms at the semantic level among processes, a business process modeling algorithm based on ontology language using concept hierarchy as background knowledge was proposed. The business process modeling by traditional business process languages can be transferred to ontology language (OWL) description so as to realize the semantic sharing of business process. The approach has two main advantages compared with traditional methods: (1) *accelerating the building process, minimizing the conversion cost*; (2) *background knowledge guides the extraction of knowledge resided in database*. As a way of validating the approach, part of the sales order process adopted from SAP reference process models was used to construct business process ontology, to validate the method. The gold standard experiments show that this method can correct and effectively construct process ontology.

Keywords

ontology, business process, concept hierarch model, process ontology

Recommended Citation

Huang Ying, Li Kangshun, Li Wei, He Keqing. Business Process Modeling Algorithm Based on Ontology Language[J]. Journal of System Simulation, 2017, 29(10): 2282-2290.

一种基于本体语言的业务流程建模算法

黄颖^{1,2}, 李康顺³, 李伟³, 何克清¹(1. 武汉大学软件工程国家重点实验, 武汉大学计算机学院, 湖北 武汉 430072;
2. 赣南师范学院数学与计算机学院, 江西 赣州 341000; 3. 华南农业大学数学与信息学院, 广东 广州 510642)

摘要: 针对业务流程在语义层缺乏信息的共享机制, 提出一种基于本体语言的业务流程建模算法, 该算法将概念层次作为背景知识, 将传统业务流程建模语言构建的业务流程转换为本体语言(OWL)描述, 从而实现业务流程的语义共享。该算法与传统的业务流程建模方法相比具有明显的优越性: (1) 加速 OWL 本体的构建, 减少转换代价; (2) 背景知识可以指导用户从数据库中提取知识。最后, 使用金本位方法, 并以 SAP 部分流程参考模型为例, 验证了提出的基于本体语言建模算法的有效性。

关键词: 本体; 业务流程; 概念层次, 流程本体化

中图分类号: TP311 文献标识码: A 文章编号: 1004-731X (2017) 10-2282-09

DOI: 10.16182/j.issn1004731x.joss.201710008

Business Process Modeling Algorithm Based on Ontology Language

Huang Ying^{1,2}, Li Kangshun³, Li Wei³, He Keqing¹(1. State Key Laboratory of Software Engineering, School of Computer, Wuhan University, Wuhan 430072, China;
2. Institute of Mathematical and Computer Sciences, Gannan Normal University, Ganzhou 341000, China;
3. College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China)

Abstract: In the view of problem in lack of information sharing mechanisms at the semantic level among processes, a business process modeling algorithm based on ontology language using concept hierarchy as background knowledge was proposed. The business process modeling by traditional business process languages can be transferred to ontology language (OWL) description so as to realize the semantic sharing of business process. The approach has two main advantages compared with traditional methods: (1) *accelerating the building process, minimizing the conversion cost*; (2) *background knowledge guides the extraction of knowledge resided in database*. As a way of validating the approach, part of the sales order process adopted from SAP reference process models was used to construct business process ontology, to validate the method. The gold standard experiments show that this method can correct and effectively construct process ontology.

Keywords: ontology; business process; concept hierarch model; process ontology

引言

语义业务流程管理^[1]是通过本体语言和语义



收稿日期: 2015-10-14 修回日期: 2016-02-28;
基金项目: 国家自然科学基金(613730376, 61573157),
国家 973 重点基础研究发展计划(2014CB340404), 江
西省自然科学基金(20142BAB217028);
作者简介: 黄颖(1981-), 女, 江西万载, 博士, 讲师,
研究方向为软件工程。

Web 服务框架来表示不同领域的企业知识, 从而提高业务流程管理的自动化水平。其目标是应用机器推理使得不同领域的企业知识可以互相理解, 特别是发现流程框架和进行流程组合。文献[1]提供了语义业务流程管理的定义, 很多学者和企业合作者在这方面开展了不少研究工作。但目前业务流程管理仍然没有在语义层提供一个统一的流程表示方

式, 其自动化程度至今仍不能让用户满意。

研究利用本体进行构建某一特定领域的知识库, 已经被人们广泛地用于解决异构环境互操作、系统工程, 以及通信等领域中的知识获取、知识共享、知识重用和系统集成等^[2]。在已有研究的基础上, 将业务流程进行本体化, 归纳总结了某一具体领域本体知识库的构建方法, 有助于设计人员高效并高质量地进行流程进一步开发利用, 用途广阔。

目前主流的业务流程语言如 BPEL(Business Process Execution Language)、EPCs(Event-driven Process Chain)、BPMN(Business Process Modeling Notation)、UML 活动图等, 它们的活动都是用自然语言描述, 所以存在语义异构性。

为了消除流程模型当中的语义异构, 流程模型可以用本体的形式进行描述, 从而支持进一步的工作。万维网联盟推荐 OWL(Web Ontology Language)作为书写本体的规范化语言 OWL 也是目前使用最广泛的本体语言, 故本文将 OWL 作为流程本体的描述语言。

本文的主要贡献是提出了一种构建业务流程本体的新算法, 将业务流程中的组件以及组件之间的关系作为构建流程本体的重要资源, 同时采用概念层次作为流程本体构建的背景知识, 该算法简称为 CH-BPO。该算法在流程本体构造过程中结合了流程的架构元素以及元素之间的关系值, 因为流程结构及其元素间包含了很多关系, 故结合概念层次作为领域知识是建立流程本体的一个很有效方法。

1 相关工作

目前提出的流程发现方法, 大多数只是基于图形的比较或者是基于 XML 的流程查询, 忽视了基于语义相似性查找。

XTR-RTO 是从 XML 文档建立 OWL 本体的方法, 转换模块将 XML 模式映射到关系数据库, 然后映射到 OWL 本体^[3]。Chishti I 及 Ma 等人提出基于时序逻辑将业务流程映射到本体映上, 该方法

支持不同流程建模语言所建立的流程模型^[4]。

参考文献[5]提出了基于本体的业务流程重组方法, 通过本体知识的协助可以帮助用户更准确地根据需求重构业务流程。文献[6]提出了一种基于本体的跨组织的协同方法, 由于本体自身具有的特点, 可协助企业准确有效的进行流程协同。

Beco 等人针对特定的语言 OWL-WS(对 workflow 和服务的 OWL)来描述工作流本体和服务^[7], 产生的工作流本体既没有关注业务流程间的关系也没有注意工作流的相似性, 这种语言主要用于特定的适应性业务流程。Zheng 等人提出了一种针对智能交通系统的 ITS 本体, 该方法根据关系数据库的内容, 提出了七种本体生成规则, 该方法主要用于智能交通领域^[8]。李亢等人提出构建数据语义集成的装备领域本体^[9]。Yahia 等人提出的方法没有采用关系数据库作为数据过渡表示形式, 只是单纯提取了 XML 文件当中的元素, 同时没有考虑元素间的相互关联, 以及数据所处的领域特点^[10]。

2 业务流程建模

2.1 业务流程图

目前常用的流程建模语言大多基于图形化形式, 甚至所谓的结构化建模语言, 如 iEPC^[11]。BPEL 本身没有图形化表示, 但它也能够映射到基于图形的概念上^[12]。同时, 这些流程建模语言之间还有重复之处^[13]: 它们都是基于活动节点, 拥有相同路由选择的节点可能有同样的属性。最后, 几乎所有的流程建模语言都可以转换为 Petri 网^[14], 其中大多数流程建模语言可以完全转换为 Petri 网, 只有少部分结构不能直接转换。因此, 如果能够正确的定义业务流程图, 它是可以表示所有的图形概念, 甚至不同图形间的差异。定义 1 是对业务流程图的形式化描述。

定义 1. 业务流程图: 设 T 为类型集合, Ω 为文本标签集合。业务流程图是一个五元组 $(V, E, \tau, \lambda, \alpha)$, 其中: V : 节点的有限集; $E: V \times V$ 是

边的有限集; $\tau: (V \cup E) \rightarrow T$ 确定节点及边的关系; $\lambda: V \rightarrow \Omega$ 确定节点的标签; $\alpha: (V \cup E) \rightarrow (T \rightarrow \Omega)$ 确定节点和边的属性, 这里属性是类型和标签的结合。

2.2 流程建模及 EPC 家族

本文使用 iEPC(Integrated EPC)作为流程建模语言。(1) 因为 EPC 的流程元素已经包含了其他流程建模语言内最常用的元素; (2) EPC 建模语言已

经有将近 20 年的工业应用历史, 被证明是切实可用的, 全球企业管理软件解决方案提供商巨头 SAP 公司也将 EPC 作为其所使用的流程建模语言之一; (3) iEPC 是对 EPC 的扩展, 它在 EPC 的基础上, 针对 EPC 流程的功能节点(functions)关联了角色(role)和对对象(object)节点, 具备了处理数据对象的能力^[11]。图 1 为订单确认业务流程的 iEPC 流程图。

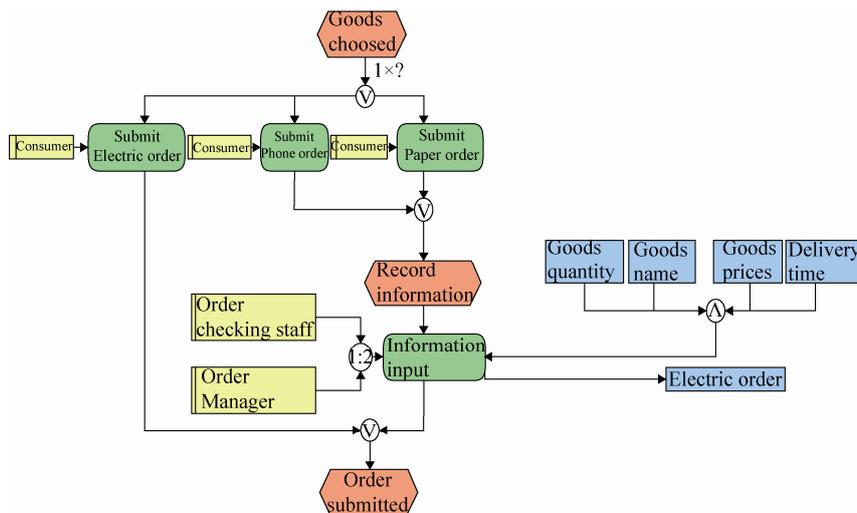


图 1 iEPC 表示的订单确认业务流程
Fig. 1 Order confirmation business process represented by iEPC

3 业务流程本体化

3.1 流程本体化框架

本体能够为特定领域和术语提供规范的表示^[15], OWL 是目前使用最广泛的一种本体语言之一。本文基于 OWL 提出了一种半自动化的流程本体构建方法(即: CH-BPO)。在流程本体构建过程中, 数据准备阶段应用了概念层次, 该阶段主要关注的是与规约有关的覆盖问题^[16], 在数据准备阶段概念层次作为有限的领域知识。领域知识表示了在本体层次中对应的层次, 它由一组属性值构成。这些信息由领域专家提供。领域知识主要提供有关领域的明确信息, 概念层次能够提供所研究领域的有关信息, 为提供何种知识提供指导。在学习阶段使用概念层次可以产生和评估假设。本文中, 假设是一种具备属性和类的概念模式, 同时

它们彼此有关联, 领域本体正符合这个要求。

图 2 描述了本文工作的构架, 从业务流程库中提取相应流程, 获取流程的特征及特征间的关系, 并把它们存储到关系数据库当中, 这是属于数据准备阶段, 特征与关系的选择依赖于领域知识。结合领域知识, 使用本体转换规则, 得到流程本体。

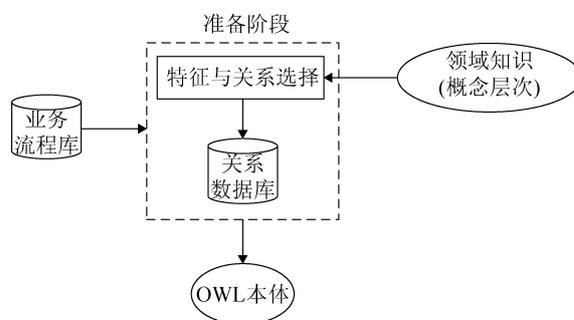


图 2 流程本体化框架图
Fig. 2 Process ontology framework

3.2 流程本体化转换规则

根据对象不同 iEPC 中节点类型可分为控制流节点和数据流节点。控制流节点包括: OR(或), AND(与), XOR(异或), Function(功能), Event(事件)节点, 数据流节点包括: Role(角色), Object(对象)和 Range(范围)节点^[11]。为了完整的表示流程及流程节点间的关系, 流程的 Edge(边)也作为本体的一种类。对 iEPC 的形式化定义见定义 2。

定义 2. 业务流程是一个三元组 $iEPC = \{Node, Property, Value\}$ $Property = \{ID, name, optional, type, lowerBound, upperBound, source, target\}$ $Node = \{E, A, F, C, R, O, C_N\}$ 。 $Node \rightarrow Property$ 。 Node 为流程当中节点的类型, Property 为流程节点的属性, Value 为节点属性的具体值。Node 与 Property 是部分映射关系。不同类型的节点属性不同, 但是它们都有一个共同的属性 ID, ID 值唯一确定每个节点。本文使用 Property.value 表示某个属性的具体值。

E: 是有限非空的事件节点集合;

A: 是有限非空的边集合;

F: 是有限非空的功能节点集合;

C: 是有限非空的连接点集合, $C = C_{OR} \cup C_{XOR} \cup C_{AND} \cup C_{RANGE}$;

R: 是有限非空的角色节点集合;

O: 是有限非空的对象节点集合。

C-iEPC 业务流程转换为本体的形式表现, 这里采用 OWL 作为本体, OWL 主要由个体(Individual)、属性(Property)和类(Class)三部分组成^[9]。个体是类的实例, 代表领域内具体的、人们实际感兴趣的那些对象。

为了便于本体语言与流程语言相对应, 对 OWL-DL 进行形式化定义, 如定义 3 所示, 本体类包含的关系见定义 4, 本体的属性模型定义见定义 5。

定义 3. OWL-DL 本体定义如下:
 $Ontology = \langle , Class, Property, Individual \rangle$,

Class 为本体类, Property 为本体的属性, Individual 为本体的实例。

定义 4. 类关系模型是二元组 $Ch = \langle C, R_C \rangle$, $R_C = \{subClassOf, intersectionOf, unionOf, complementOf, Restriction, oneOf, H_{C_i}\}$ 这里 C 是一个有限非空的类的集合, H_{C_i} 表示类所处的层次, $i=1, 2$ 。 R_C 表示类 C 的特定关系。

定义 5. 属性关系模型是一个四元组, $Ph = \langle OP, DP, R_{OP}, R_{DP} \rangle$, 这里 Ph 是一个有限非空的属性集合, OP 为对象属性(Object property), DP 为数据属性(Data property), $R_{OP} = \{subPropertyOf, domain, range, disjointWith, equivalentProperty, FunctionalProperty, TransitiveProperty, ReflexiveProperty, IrreflexiveProperty, SymmetricProperty, InverseFunctionalProperty, AsymmetricProperty, H_{OP_i}\}$ H_{OP_i} 为对象属性所处的层次, R_{OP} 是关于对象属性的特定关系。

$R_{DP} = \{subPropertyOf, range, disjointWith, equivalentProperty, FunctionalProperty, H_{DP_i}\}$, H_{DP_i} 为数据属性所处的层次, $i=1,2$, R_{DP} 是关于数据属性的特定关系。

完成了流程及本体的定义之后, 可以根据定义 6 对二者进行映射。

定义 6. 流程节点与 OWL-DL 本体的映射关系 ($F(BP) \rightarrow Ontology$) 定义如下:

$\Phi = C_N$ 获取节点;

$\psi = Property \setminus \{ID\}$: 获取除 ID 之外的属性值;

$L_{id} = C_N \rightarrow (Property.ID)$: 获取节点对应的 ID 属性;

$\zeta = C_N \rightarrow \psi$: 获取节点的除 ID 之外对应的属性;

$G_{name} = \Phi \rightarrow (Property.name)$: 获取节点对应的 name 属性;

$G_{id} = \{C\} \rightarrow (Property.id)$: 获取连接点对应的 id 属性;

$V(G_{name}) = name.value$: 获取流程节点属性

集合成员的 name 值;

$V(G_{id}) = id.value$: 获取流程节点属性集合成员的 id 值;

$V(L_{id}) = id.value$: 获取流程可变连接点属性集合成员的 id 值;

$F(C_N) \rightarrow H_{C_1}$: 将业务流程的节点类型映射到本体的一级类;

$V(G_{name}) \rightarrow H_{C_2}$: 将业务流程的流程属性中 name 属性的具体值映射为本体的二级类;

$F(C_N) \rightarrow H_{DP_1}$: 将业务流程的节点类型映射为本体的一级数据属性;

$V(G_{name}) \rightarrow H_{DP_2}$: 将业务流程中连接点的节点属性中 name 属性映射为本体的二级数据属性;

$V(G_{id}) \rightarrow H_{DP_2}$: 将业务流程的连接点的节点

属性中 ID 属性映射为本体的二级数据属性;

$V(L_{id}) \rightarrow I$: 将业务流程的节点属性中的 ID 属性的值映射为本体的实例;

$V(\zeta) \rightarrow H_{DP_2}.value$: 将业务流程的节点属性中除 ID 属性之外的属性的值映射为本体数据属性的值。

首先构建本体的类, 然后为它添加本体的属性。本体有两大类属性: 数据属性和对象属性。数据属性有三类元素: ID 号、属性定义域和属性值域。根据流程的特点, 将流程节点的 ID 值作为其数据属性的 ID 号。流程本体的数据属性的定义域是其所属的类。值域是数据属性的数据类型。流程本体的对象属性是两个流程类之间的关系。

3.3 概念层次作为领域知识

领域知识是要探索领域的信息, 对知识发现很有用。概念层次是领域知识的一种表达形式, 概念层次可以在多个抽象层次上发现知识。概念层次由节点集合构成, 每个节点代表一个概念。图 3 为本文所使用的概念层次, 它可以在数据准备阶段为生成流程本体提供指导。

数据准备阶段的输入为流程集合以及与流程领域相对应的概念层次。概念层次是由领域专家建立

的, 它的值会随着领域和流程建模语言的不同而有所变化。对流程本体化限制的概念层次如下所示:

$\forall (pv_i, l_i) | (pv_i \in PV) \wedge (l_i \in L) \wedge (\bigcup_{1 \leq i \leq n} (pv_i, l_i) = 0)$, $PV = (pv_1, pv_2, \dots, pv_n)$ 是属性集合, $L = (l_1, l_2, \dots, l_n)$ 层次集合, (pv_i, l_i) 是属性集的值 pv 及其它们对应的层次 1, (PV, L) 是 PV 与 L 的集合对, $\bigcap_{1 \leq i \leq n} (pv_i, l_i) = 0$ 表示任意一组属性-层次对与其他属性-层次对都是不同的。同样, 可以说任意属性值与其对应的层次都不同。

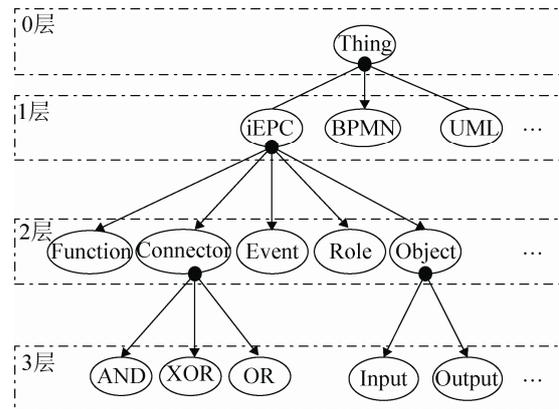


图 3 流程本体化概念层次
Fig. 3 Concept hierarchy of process ontology

3.4 流程本体化算法

算法 1. 基于概念层次流程本体化算法

输入: 流程集合 PC, 概念层次 CH

输出: 流程本体 PO

- 1 ProcessSchema.name ← P.name
- 2 FOR level=0 TO level < numofCH
- 3 IF ∃ concept IN CH mapping P
- 4 THEN item ← concept
- 5 Table.name ← item
- 6 Table.column ← item.pt
- 7 Table.data ← item.data
- 8 ∇ ProcessSchema IN Database
- 9 PO.name ← Schema.name
- 10 PO.class ← Table.name
- 11 PO.OPV ← tableRelation
- 12 PO.FirstDP ← Table.name
- 13 PO.SecondDP ← Table.column
- 14 PO.Individual ← item.data.ID
- 15 PO.IndividualDPV ← item.data.Others
- 16 RETURN PO

算法首先将流程名称设为数据库的名称(1行); 针对概念层次, 从第 0 层到最后逐层扫描, 在流程当中寻找是否有元素与概念层次中的概念相匹配(2~3 行); 如果有匹配的项(item), 则将 item 及其相关数据存储在数据表当中, 如 item 名为数据表名, item 属性为数据表的列名, item 中的数据为数据表的数据值(4~7 行)。然后对数据库中的所有流程数据模式(8 行), 将数据模式名为流程本体名(9 行), 数据模式中的数据表名为流程本体类名(10 行)及一阶数据属性(12 行), 数据表中的其他值为流程本体的对象属性(11 行), 数据表的列名为流程本体二阶数据属性(13 行), 数据表中对应的 ID 值为流程本体的实例名(14 行), 数据表中的其他值为流程本体实例的数据属性值(15 行)。最终返回流程本体(16 行)。

算法由两大部分构成, 第一部分是概念层次与业务流程相映射, 如果在概念层次当中能够找到与之对应的流程元素则将流程当中的元素及其属性存入数据库当中; 第二部分将数据库中的元素生成流程本体。不论是哪种类型的业务流程, 其元素的类型种类是有限的完全能够在线性时间内完成与流程元素类型相关的操作, 真正影响算法效率的是流程的当中的节点数, 本文设节点数为 n , 故算法的时间复杂度为 $O(n)$ 。

4 仿真实验

4.1 实验准备工作

实验以 iEPC 为流程建模语言, 其他流程建模语言可以用同样的方法进行本体转换。实验过程中, 为了灵活处理流程元素, 将流程的元素存储在数据库中, 然后从数据库中抽取所需的元素及其相关属性, 将其转换为 OWL 本体。

根据业务流程当中各种节点的类型, 将其存入数据库当中, 数据库名为流程的名称, 数据表名为节点的类型(如: role), 数据表中的列名为每种节点中包含的属性。如可变的角色节点的属性就包含: ID(标号), Name(名称)和 Optional(可选

的)。数据表中的具体数据即为对应节点的属性的具体值。具体设置如表 1 所示。

表 1 iEPC 转化为 MySQL 数据集的规则
Tab. 1 Rules of iEPC transfers to MySQL data set

iEPC	MySQL 数据库
Epml 文件	数据库名
节点类型	表名
节点属性	列名
节点属性值	数据项值

在业务流程的节点存入数据库之后, 这些数据就成为构建本体的有价值的资源, 使用模式转换可以将关系数据库中的知识转用 OWL 本体来表示。

使用模式转换将关系数据库中的数据和数据间的关系转换为 OWL 本体的数据、数据属性及数据间的关联, 具体实现步骤如下:

首先, 将数据库名设置为 OWL 本体名, 这并不是 OWL 本体的 URI, 不能用这个名称在互联网上发布所生成的本体, 生成的流程本体的 URI 是指定或者随机生成的一个可通过 HTTP 协议访问的资源。

第二, OWL 本体中的类名来自数据库中的数据表名。

第三, 在添加完 OWL 本体类之后, 添加本体的属性。本体属性分为数据属性和对象属性。OWL 本体的数据属性有三个要素: 一是属性的标识值, 这个可以用流程节点属性的 ID 号或者属性名表示, 为了便于查看设立两级本体数据属性, 一级本体数据属性名为数据表的表名, 二级本体数据属性名为数据表的列名; 二是数据属性的定义域, 定义域指的是数据属性所属的类; 三是数据属性的值域, 值域指的是数据属性的数据类型。对象属性是表示两个类之间的关系, 本文将两个数据表间的关系视为对象属性。

最后, 将数据库中的数据项转换为业务流程节点本体的个体, 但是仅用每个数据表中的 ID 项表示对应的个体, 因为在 iEPC 中所有元素的 ID

号都是唯一的。数据表中的其他数据项作为对应个体的本体数据属性的值。具体转换规则如表 2 所示。

表 2 MySQL 数据项与 OWL 本体转换的规则
Tab. 2 Rules of MySQL data items transfer to OWL

数据库	本体
数据库名	OWL 本体名
数据表名	OWL 本体的类名
数据表的列名	OWL 本体一级数据属性名
	OWL 本体二级数据属性名
数据项 ID 号	OWL 本体的个体名
其他	与个体对应的数据属性的值

4.2 流程本体化实验

实验使用金本位(gold standard)本体来评估构建的流程本体的质量。自然语言处理系统的带标语料库被业内人士誉为金本位，因为它们常用作训练和测试的语料，起到标准答案的作用。本文金本位本体是指经过人工标注或审定的带标本体库，我们将本体转化之后的流程本体与金本位本体进行对比，这是对所考虑问题领域中概念常用的判定方法^[9]。

文献[18]介绍了对本体的准确率和召回率的计算方法。给定一个计算核心本体(OC)和一个参考本体(OR)，本体的准确率和召回率定义如下：

$$LP(O_C, O_R) = \frac{|C_C \cap C_R|}{|C_C|},$$

$$LR(O_C, O_R) = \frac{|C_C \cap C_R|}{|C_R|},$$

$$LP(O_C, O_R) = LR(O_R, O_C)。$$

评价和比较本体使用最多的表示是准确率和召回率^[9]，本文使用分类准确率(Taxonomic Precision:TP)和分类召回率(Taxonomic Recall:TR)，因为在构建本体的时候存在对属性和类命名不适宜的问题，分类准确率和分类召回率能较好的解决这个问题。本体分类准确率和分类召回率的计算是基于语义覆盖(Semantic Cotopy:SC)，它定义了本体(O)中所有的父概念和子概念(c)之间的关系：

$$SC(c, O) = \{c_i | c_i \in C \wedge ((c_i \leq c) \vee (c_i \geq c))\}$$

公共语义覆盖(Common Semantic Cotopy: CSC)可以避免词汇准确率对分类的影响，它摒弃了不在其中的概念集。公共语义覆盖定义如下：

$$CSC(c_i, O_1, O_2) = \{c_j | c_i \in C_1 \cap C_2 \wedge (c_j \leq C_1.c_i)\}$$

假设一个特征提取 ce，两个概念 c₁ ∈ O_C 和 c₂ ∈ O_R 的局部分类准确率 tp_{ce} 定义如下：

$$tp_{ce}(c_1, c_2, O_C, O_R) = \frac{|ce(c_1, O_C) \cap ce(c_2, O_R)|}{|ce(c_1, O_C)|}$$

公共语义覆盖的准确率 TP_{CSC} 和召回率 TP_{CSC} 可以用以下公式计算：

$$TP_{CSC}(O_C, O_R) = \begin{cases} \frac{1}{|C_C|} \sum_{c \in C_C} tp(c_1, c_2, O_C, O_R) & \text{if } c \in C_C \\ 0 & \text{if } c \notin C_C \end{cases}$$

$$TR_{CSC}(O_C, O_R) = TP_{CSC}(O_R, O_C), C_c \text{ 是概念集。}$$

和信息检索领域的准确率和召回率一样，分类准确率和召回率之间也需要一个平衡，所以也要分类的 F-measure，它是全局分类准确率和召回率的一个调和手段：

$$TF(O_C, O_R) = \frac{2 \cdot TP(O_C, O_R) \cdot TR(O_C, O_R)}{TP(O_C, O_R) + TR(O_C, O_R)}$$

如果 TF 不受词汇层次影响，可以使用分类 F'-measure^[20]，它是对 LR 和 TF 的一个调和手段

$$TF'(O_C, O_R) = \frac{2 \cdot LR(O_C, O_R) \cdot TF(O_C, O_R)}{LR(O_C, O_R) + TF(O_C, O_R)}$$

本文以物流配送流程为例，将其进行本体化。图 4 是图 1 对应流程的本体化结果。

所有实验均基于如下环境：处理器 Intel(R) Core(TM) i5, 2.6GHz, 8GB 内存, OS X 64 位操作系统, JAVA JDK1.7。本文选取了 SAP 的 604 个参考业务流程模型当中 30 个销售订单流程参考模型为实验流程。SAP 参考业务流程模型是业务流程领域内广泛使用的具有代表性的数据集，国内外很多学术论文都是基于该数据集进行试验证明的，故本文也采用 SAP 参考业务流程模型作为试验对象。SAP 参考业务流程模型中的业务流程均是以 EPC 语言建模的。Xu 等人提出了从关系数据库中提取 OWL 本体的方法^[3]，Thuan 利用众包方法建

立企业业务流程本体^[21], 本文提出方法是经由关系数据库生成的业务流程本体, 故采用基于语义 cotopy 的方法将本文方法与上述两种本体生成方法

进行比较。表 3 是 CH-BPO 与 XTR-RTO^[3]及参考文献[21]提出的方法(本文称其为: TXJJ)对语义覆盖评估对比实验的结果, 表中显示的是平均值。

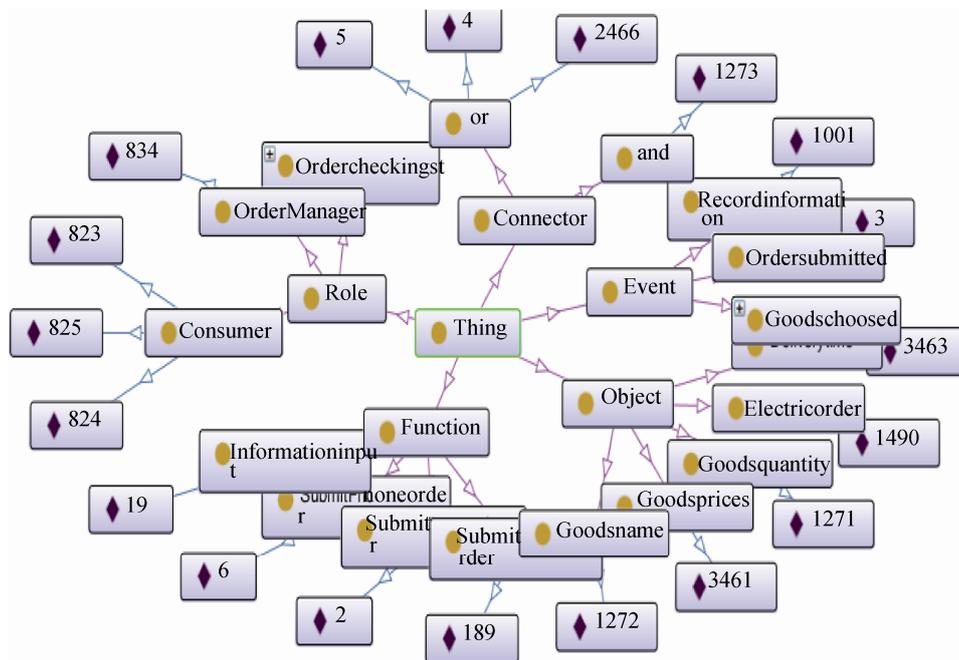


图 4 物流流程本体

Fig. 4 Ontology of logistics process

表 3 语义 cotopy 的评价结果

Tab. 3 Evaluation result of semantic cotopy %

Method	LP	LR	TP _{csc}	TR _{csc}	TF _{csc}	TF' _{csc}
CH-BPO	87	94	87	92	89	92
XTR-RTO	85	93	84	93	88	90
TXJJ	79	94	76	92	83	88

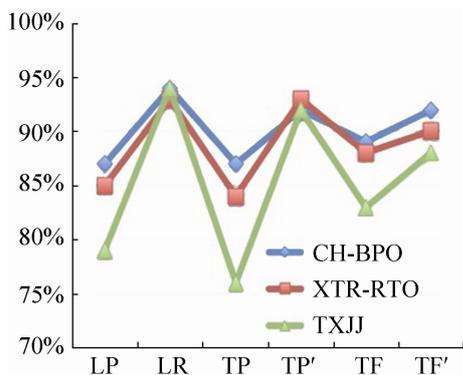


图 5 实验效率对比图

Fig. 5 Comparison of experimental efficiency

从表 3 和图 5 可以看出 CH-BPO 的 TF_{csc}、TF'_{csc}、比其他两种算法都更高, 准确率也有了

一定的提高, 召回率基本相似。总体结果优于其他两种方法。

5 结论

语义标注是实现异构资源语义互操作的手段之一。由于目前存在多种流程建模语言, 有必要对这些流程模型的语义异构性进行管理。本文提出一种从业务流程建模语言中提取本体的新算法, 该算法从业务流程提取元数据然后建立对应流程本体。在本体建立过程中, 根据流程的特点, 选择流程的节点类型作为特征, 在数据准备阶段结合概念层次帮助识别和选择相关数据。

在建立流程本体的过程中结合概念层次作为领域知识, 该方法有两大优点: (1)加速了 OWL 本体的构建, 减少了转换代价; (2)领域知识指导用户从数据库中提取所需的知识, 从而更好的实现流程本体构建。实验证明本文提出的算法比单纯使用数据映射进行流程本体化转换更加有效。

参考文献:

- [1] Hepp, M., Leymann, F., Domingue, J., et al. Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management [C]// IEEE International Conference on e-Business Engineering. USA: IEEE, 2005: 535-540.
- [2] Ostrowski L, Helfert M, Gama N. Ontology engineering step in design science research methodology: a technique to gather and reuse knowledge [J]. Behavior & Information Technology(S0144-929X), 2014, 33(5): 443-451.
- [3] J Xu, W Li. Using Relational Database to Build OWL Ontology from XML Data Sources [C]// Proceeding 2007 International Conference on Computational Intelligence and Security Workshops. USA: IEEE, 2007: 124-127.
- [4] Chishti I, Ma J, Knight B. Ontology Mapping of Business Process Modeling Based on Formal Temporal Logic [J]. International Journal of Advanced Computer Science and Applications (S2158-107X), 2014, 5(7): 95-104.
- [5] Rao L, Mansingh G, Osei-Bryson K M. Building ontology based knowledge maps to assist business process re-engineering [J]. Decision Support Systems (S0167-9236), 2012, 52(3): 577-589.
- [6] Hoang H H, Jung J J, Tran C P. Ontology-based approaches for cross-enterprise collaboration: a literature review on semantic business process management [J]. Enterprise Information Systems (S1751-7575), 2014, 8(6): 648-664.
- [7] Stefano Beco, Barbara Cantalupo, Ludovico Giammarino, et al. OWL-WS: A Workflow Ontology for Dynamic Grid Service Composition[C]// International Conference on E-Science and Grid Computing. IEEE Xplore, 2005: 148-155.
- [8] Zheng K, Zhu D, Yan L, et al. A model of extracting ITS ontology from a relational database [C]// 14th COTA International Conference of Transportation Professionals: Safe, Smart, and Sustainable Multimodal Transportation Systems. China (ASCE), 2014: 258-267.
- [9] 李亢, 李新明, 刘东. 面向数据语义集成的装备领域本体构建研究 [J]. 系统仿真学报, 2015, 27(5): 1071-1080. (Li Kang, Li Xinming, Liu Dong. Study on equipment domain ontology construction for integration of data semantics [J]. Journal of System Simulation (S1004-731X), 2015, 27(5): 1071-1080.)
- [10] Yahia N, Mokhtar S A, Ahmed A W. Automatic Generation of OWL Ontology from XML Data Source [J]. International Journal of Computer Science Issues (S1694-0814), 2012, 9(2): 77-83.
- [11] La Rosa M, Dumas M, TerHofstede A H M, et al. Configurable multi-perspective business process models [J]. Information Systems (S0306-4379), 2011, 36(2): 313-340.
- [12] Kuster J, Ryndina K, Gall H. Generation of business process models for object life cycle compliance [C]// Proc. of BPM, LNCS. Germany: Springer, 2007, vol. 4714: 165-181.
- [13] Mens T. A state-of-the-art survey on software merging [J]. IEEE Transactions on Software Engineering (0098-5589), 2002, 28(5): 449-462.
- [14] Ohst D, Welle M, Kelter U. Differences between versions of UML diagrams [C]// Proc. of ESEC/SIGSOFT FSE. USA: ACM, 2003: 227-236.
- [15] Sowa J F. Knowledge representation: logical, philosophical and computational foundations [M]. Vol. 13. Pacific Grove: Brooks/Cole Publishing Co. 2000.
- [16] W3C OWL Working Group. OWL 2 Web Ontology Language Document Overview (Second Edition) [Z], (2012-11-11)[2015-03-23] <http://www.w3.org/TR/owl2-overview/>.
- [17] Dellschaft K, Staab S. On how to perform a gold standard based evaluation of ontology learning [C]// The Semantic Web-ISWC 2006. Berlin, Heidelberg, Germany: Springer 2006: 228-241.
- [18] Sabou M, Wroe C, Goble C, et al. Learning domain ontologies for web service descriptions: an experiment in bioinformatics [C]// Proceedings of the 14th international conference on World Wide Web. USA: ACM, 2005: 190-198.
- [19] T Zhang, D Xu, J Chen. Application-oriented purely semantic precision and recall for ontology mapping evaluation [J]. Knowledge-based Systems (S0950-7051), 2008, 21(8): 794-799.
- [20] Cimiano P, Hotho A, Staab S. Learning concept hierarchies from text corpora using formal concept analysis [M]. AI Access Foundation, 2005: 305-339.
- [21] Thuan N H, Antunes P, Johnstone D, et al. Building an Enterprise Ontology of Business Process Crowdsourcing: A Design Science Approach [C]// PACIS 2015 Proceedings. Singapore (AISel), 2015.