

跨组织 workflow 集成中间件语义转换部件研究

张月菊, 王 涛, 林 拉

(华南师范大学 计算机学院, 广东 广州 510631)

摘 要: 跨企业的系统协同工作是目前企业信息化的难点之一。提出一种新的跨组织功能集成实现方案——领域语义控制转换, 对其核心部件语义存储及转换模块给出了三种模型, 并对其进行比较, 选择其中较优模型——推荐式领域语义控制转换模型。该模型包括: 语义信息库、语义推荐、语义存取、语义信息检索和子系统控制模块, 以完成系统间应用、功能数据集的语义匹配和控制工作。最后给出了简要的实现方案和实例, 通过语义转换部件来实现跨组织功能的集成。

关键词: 计算机协同工作; 工作流; 语义控制转换; 语义映射

中图分类号: TP31

文献标识码: A

文章编号: 1673-629X(2010)03-0092-04

Research on Structure for Semantic Translation in Cross-Organizational Workflow Integrated System

ZHANG Yue-ju, WANG Tao, LIN La

(School of Computer Science, South China Normal University, Guangzhou 510631, China)

Abstract: "Cross-enterprise system cooperation" is a current difficulty of enterprise informatization. A new approach to realize cross-organization process integration - domain semantic translation - is proposed, three models of its core modules to implement semantic storage, translation and mapping are introduced and compared, from which a preferable model called "recommending model" is chosen. This model includes semantic information repository, semantic recommendation module, semantic access module, semantic information search module, sub-systems control module, to implement cross-system matching and controlling on data semantic and function semantic. A brief implementation scheme and sample are introduced at last. The integration is realized.

Key words: CSCW; workflow; semantic translation; semantic mapping

0 引言

早在 1984 年美国 MIT 的 Irene Greif 和 DEC 的 Paul Cashman 就提出了 CSCW (计算机协同工作), 用于描述他们正在组织安排的、有关如何用计算机支持交叉学科的人们共同工作的课题^[1]。1993 年工作流管理联盟 WfMC 的成立也标志着工作流技术在计算机应用领域取得自己的一席之地^[2]。

工作流管理系统 WfMS 为改善流程效率提供了计算机协同支持环境, 它在企业内部得到了成功应用^[3]。但是, 随着 Internet 技术的发展, 工作流管理系统从协调企业内部的各种资源转移到着重处理系统间的资源协调和互操作。

由于不同的企业内部往往部署来自不同厂商的工作流管理系统, 由于不同的工作流管理系统没有采用统一的标准所带来的异构性, 如采用的底层通信基础结构各不相同, 不同的厂商工作流产品具有自己独立的工作流模型、工作流定义语言以及 API 函数, 标准化程度差, 不同的工作流管理系统之间很难实现互操作^[4]。

实际上, 要求不同的企业采用同种工作流系统是不现实的, 如何集成各种遗留系统和系统之间的互操作是解决问题的关键。实现互操作的根本途径是基于语义的信息交换^[5], 而现有的工作流系统很少是基于相同的语义实现的, 不同系统间的语义如何能被相互识别, 这就需要用到语义映射技术。因此, 构建语义控制转换模型, 通过语义映射完成不同系统间的语义匹配和转换, 是实现系统互操作的基础。

基于上述原因, 提出了三种语义控制转换模型并分析讨论, 找到一种最优模型——推荐式领域语义控制转换模型, 给出了较为详细的实现方案。

收稿日期: 2009-07-08; 修回日期: 2009-10-14

基金项目: 广东省自然科学基金资助项目(06300905); 广东省科技计划资助项目(2007B010200069)

作者简介: 张月菊(1980-), 女, 河北晋州人, 硕士研究生, 研究方向为工作流; 王 涛, 副教授, 博士, 研究方向为软件工程和形式化方法; 林 拉, 副教授, 研究方向为软件工程。

1 定义

1.1 语义互操作定义

IEEE 的 Standard Upper Ontology Working Group (SUO 工作组)关于语义互操作的严格定义如下:

系统 R(发送者)和系统 S(接收者)之间具有语义互操作性;而且,对于 R 和 S 之间传递的数据 p 同时满足如下条件:

对于系统 S 中的数据 P 所蕴含的每个表述 q,在系统 R 中有一个相应的表述 q',并且:

- ① q' 是系统 R 中 p 所蕴涵的;
- ② q' 和 q 在逻辑意义上是相等的。

更通俗和更容易理解的定义可以表述如下:

语义互操作是指计算机能够通过知识体系的参照、映射或其他方法,理解多个领域的知识表达,并使信息系统具有语义交互的能力。

1.2 语义映射机制

不同的 workflow 系统往往是基于不同的语义实现的,在系统间互操作时,需要由语义映射来实现语义的等价表达,以解决由于语义不能被识别造成的无法通信^[6]。

语义映射机制实质上是在不同的语义对象之间建立一种映射关系函数^[7],记为 M。即通过语义转换规则集 R 完成源语义对象 O_s 到目标语义对象 O_t 的映射,记为 $M(O_s, O_t)$ 。

2 语义控制转换模型及比较

系统间协同工作(即语义互操作)的难点在于,不同的系统并不是基于相同的语义实现的,系统间进行通信时,系统 R 的语义不能被系统 S 识别。因此,建立完备的领域语义信息库,是系统协同工作的前提。

设计和实现领域语义控制转换器,实现系统间的语义匹配和转换,系统协同工作才能得以进行。语义控制转换器的实现原理即通过语义映射完成系统间的语义匹配工作,并将源系统 R 转换后的应用数据集和功能数据集通过语义控制模块提交给目标系统 S。

2.1 语义控制转换模型

在这里,提出了三种领域语义控制转换模型。

2.1.1 集中式领域语义控制转换器

集中式领域语义控制转换器的设计思路:

如图 1 所示,语义信息库、语义信息的维护和匹配工作都集中在语义转换控制器端(语义控制转换器独自占用一台服务器,各系统的语义控制转换都集中在这台服务器上处理)进行处理。当子系统 1 和子系统 2 发生交互时,子系统 1 首先通过语义控制转换器,将系统中的应用和功能数据集 q 通过语义映射转化为子

系统 2 中的应用和功能数据集 q',等待子系统 2 完成子系统 1 的请求,操作完成后,通过语义匹配转化为子系统 1 可以识别的数据集并返回给子系统 1。

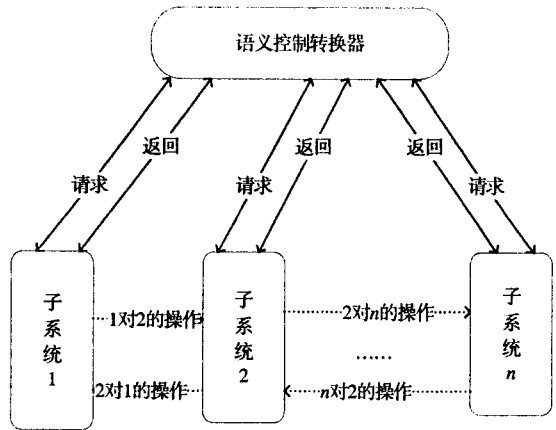


图 1 集中式领域语义控制转换模型

2.1.2 分散式领域语义控制转换器

分散式领域语义控制转换器的设计思路如下:

如图 2 所示,语义信息库、语义信息的维护和匹配工作在各系统的控制端进行处理。当子系统 1 和子系统 2 发生交互时,子系统 1 首先通过本系统的控制端,将本系统中的应用和功能数据集 q 通过语义映射转化为子系统 2 中的应用和功能数据集 q',并等待子系统 2 完成子系统 1 的请求,再次通过语义匹配转化为子系统 1 需要的数据集。

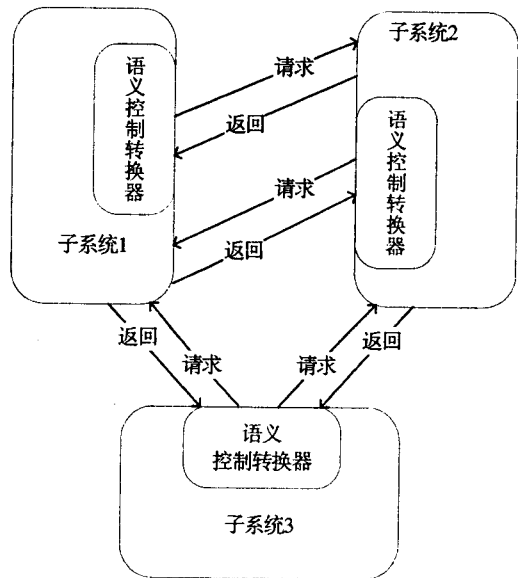


图 2 分散式领域语义控制转换模型

2.1.3 推荐式领域语义控制转换器

推荐式领域语义控制转换器的设计思路如下:

如图 3 所示,结合集中式和分散式语义控制转换的设计思路,将语义信息库、语义信息的维护和匹配工作都放在一台服务器上,将发现语义这个工作放在各子系统。一是子系统通过将语义信息库中不存在的语

义信息推荐给服务器上的语义控制转换器,由语义控制转换器来决定是否要将其扩充到语义信息库中;二是由语义控制转换器直接添加新的语义信息;三当子系统 1 和子系统 2 发生交互时,子系统 1 首先通过语义控制转换器,检索子系统 1 中的应用和功能数据集 q 是否存在语义映射,若不存在,通过语义推荐模块提交给语义控制转换器,若存在,通过语义映射转化为子系统 2 中的应用和功能数据集 q' ,等待子系统 2 完成子系统 1 的请求,操作完成后,通过语义匹配转化为子系统 1 可以识别的数据集并返回给子系统 1。

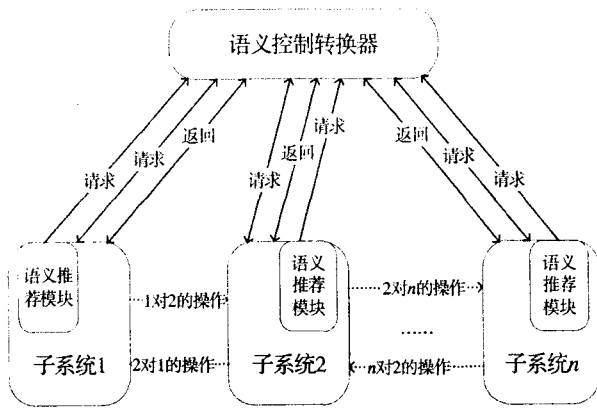


图 3 推荐式领域语义转换控制模型

2.2 语义控制转换模型比较

2.2.1 集中式领域语义控制转换器

(1)优点:

① 所有的语义都存储在语义控制转换器端的语义信息库,语义匹配工作都在语义控制转换器端执行。系统 R 仅需要提交系统交互操作所需要的数据和行为,然后则由语义控制转换器负责通知系统 S 可以识别的数据和行为。

② 便于统一管理,不会出现由于局部更新造成语义间匹配不一致。

③ 权限可以严格得到控制,只有语义控制转换器端的管理员才有维护权限。

(2)缺点:

① 语义控制转换器负载大,语义信息更新缓慢,语义信息维护的工作任务繁重。

② 各终端新出现的语义信息不能得到及时更新。

2.2.2 分散式领域语义控制转换器

(1)优点:

④ 各自要用到的语义及语义关系存储在本地数据库中,当有新的语义出现时,可以做到及时更新,并且都存储在本地,查询及传输都很迅速。

(2)缺点:

① 各自维护一个数据库,会造成数据冗余。

② 集成各个数据库时可能会出现数据不一致的情况。

2.2.3 推荐式领域语义控制转换器

优点:

① 将语义信息的搜寻扩大到各个系统,而将语义的添加维护集中在语义控制转换器。

② 同时具备了分散式和集中式的优点。

③ 避免了数据冗余和数据可能会出现不一致的缺点。

推荐式领域语义控制转换器结合了集中式和分散式语义转换控制器优点,摒弃了两者的缺点,因此,采用推荐式领域语义控制转换器来完成语义的匹配转换工作。

3 语义控制转换器的结构

领域语义转换控制器主要由 5 大部分构成,它们分别是领域语义信息库、语义推荐模块、语义存取模块、语义信息检索模块和子系统控制模块。结构如图 4 所示。

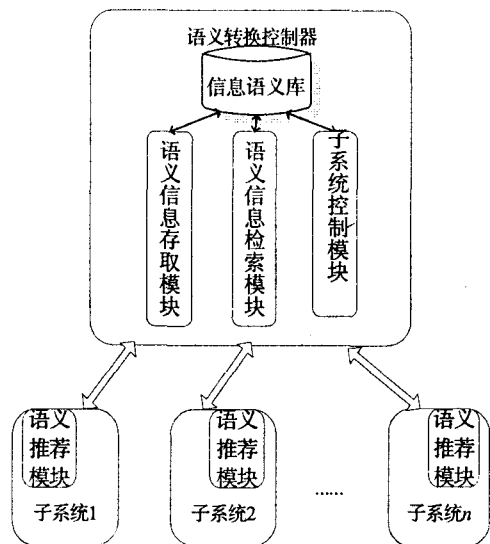


图 4 语义控制转换器的结构

3.1 领域语义信息库

存储日益增加的某领域的语义信息、语义映射函数和语义转换规则集。

3.2 语义推荐模块

该模块负责对各系统新出现的语义信息进行处理。当某个系统出现语义信息库中未包含的语义时,由附加在系统上的语义推荐模块推荐给服务器端的语义控制转换器,将其交由服务器端的语义控制转换器处理,并将处理结果反馈给该系统。

3.3 语义存取模块

语义存取模块为用户提供了操作界面,使得非领

域专业的用户也可以帮助创建和丰富该领域内的语义信息库。所维护的领域语义信息是 OWL 层次的关系。

3.4 语义信息检索模块

系统 R 和系统 S 发生互操作时,由系统 R 去访问语义控制转换器的语义信息检索模块并返回相应的数据结果集以完成系统 R 和系统 S 之间的互操作。

3.5 子系统控制模块

将模块 4 中的数据结果集合交由子系统控制模块,由该模块负责通知目标系统 S 所需要的数据和要执行的操作,等目标系统完成要做的操作之后,再将处理结果返回给源系统 R。

4 主要技术特点

对知识本体进行形式化的语言有很多,如 DAML + OIL、OWL 等。

当前有很多种不同的本体语言,它们各有千秋,如 SHOE(Simple HTML Ontology Extensions)、XOL(XML - based Ontology exchange Language)、OML(Ontology Markup Language)、CKML(Conceptual Knowledge Language)、OIL(Ontology Interface Layer)、DAML(DARPA Agent Markup Language) + OIL、OWL(Web Ontology Language)。

其中 OWL 是在总结了其他许多本体语言经验的基础上形成的,而 W3C(World Wide Web Consortium)目前的最新标准是 OWL,所以 OWL 很有可能成为将来最流行的本体描述语言。OWL 能够被用于清晰地表达词汇表中的词条(term)的含义以及这些词条之间的关系。而这种对词条和它们之间的关系表达就称作 Ontology。OWL 相对 XML、RDF 和 RDF/Schema 拥有更多的机制来表达语义,从而 OWL 超越了 XML、RDF 和 RDF/Schema 仅仅能够表达网上机器可读的文档内容的能力。可以把 OWL 分为三个子语言:OWL - Lite、OWL - DL 和 OWL - Full,主要的分类依据就是它们的表达能力。其中,OWL - Lite 是表达能力最弱的子语言,OWL - Full 具有最强的表达能力,而 OWL - DL 的表达能力则在它们之间。可以认为 OWL - DL 是 OWL - Lite 的扩展,而 OWL - Full 是 OWL - DL 的扩展^[8]。

因此,采用 OWL 来描述语义。下面给出了文中使用 OWL 描述语义的实例。

例如,假定在中间件的企业接口端的数据为:企业的支付人、保险公司的投保人;在中间件的银行端的数据为:转帐。

企业的支付人、保险公司的投保人在做转帐处理

时,对应着银行系统中的转帐,即在此中间件中,它们有相同的语义。

下面的代码是本体中一个类的描述,该类描述了一个 Payor 类及其所拥有的属性和映射关系,如图 5 所示。

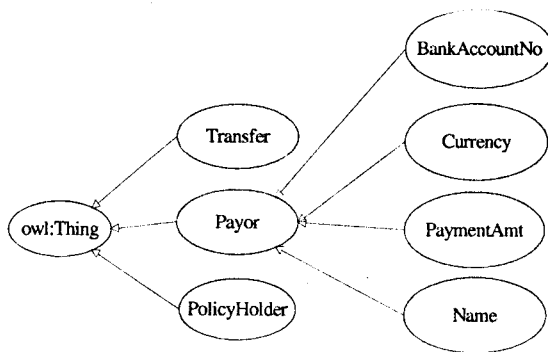


图 5 本体中类的描述

```

/* 定义支付人类 */
/* 在这一部分中,将转出方和支付人中相同的属性从转出方中引入 */
<owl:Class rdf:ID="Payor"/>
<owl:DatatypeProperty rdf:ID="BankAccountNo"> /* 付款帐号 */
<owl:equivalentClass rdf:resource="&Transfer;AccountNo"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Currency"> /* 付款币种 */
<owl:equivalentClass rdf:resource="&Transfer;Currency"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="PaymentAmt"> /* 支付金额 */
<owl:equivalentClass rdf:resource="&Transfer;TransactionAmt"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Name"> /* 支付人姓名 */
<owl:equivalentClass rdf:resource="&Transfer;Transactor"/>
</owl:DatatypeProperty>
    
```

由上面的代码可以看出,“paymentAmt”属性和 Transfer 类中的“TransactionAmt”属性等价,“Name”属性和 Transfer 中的“Transactor”属性等价。

5 结束语

系统间协同工作的基础是有可以相互识别的语义,如何处理好语义的匹配工作以及系统间的通信,文中给出了三种实现方案。从实现原理、过程和结果等方面做了对比,找到一种最优的实现方案 - 推荐式领域语义控制转换模型,之后就这种模型给出了实现的框架和技术。

证明:与引理 3 的证明完全相同。

定理 3 设 $n \geq 9$, 则 $|A(2)| \leq (n-1) \times 2^{-(n-2)}$ 。

证明:由表 6 知:串“000”与“001”,“010”与“011”,“100”与“101”,“110”与“111”分别具有相同的相关串结构,所以由引理 10 知它们长度不超过 n 的相关串的个数分别相等,且它们长度大于 n 的相关串的个数的下界为 0。若设比特串“000”,“011”,“100”,“110”的长度大于 n 的相关串的个数之和的上界为 M , 则 $|A(2)| \leq 2^{-(n-1)}M$, 由引理 9 和引理 12 知: $M = \sum_{k=4}^5 C_{n-k-2-1}^{2-1} + 3 \times 2 + 1 \times 3 + 2 \times 2 = 2n - 2$ 。

所以 $|A(2)| \leq (n-1) \times 2^{-(n-2)}$ 。

定理 4 $\lim_{n \rightarrow \infty} A(k) = 0$ 。

证明:由定义 4 知: $A(k) = 2^{-(n-1)}(N(k,00) - N(k,01) - N(k,10) + N(k,11)) = 2^{-(n-1)}(N_1(k,00) + N_2(k,00) - N_1(k,01) - N_2(k,01) - N_1(k,10) - N_2(k,10) + N_1(k,11) + N_2(k,11))$ 。由引理 10 知,比特串“00”与“01”,“10”与“11”分别具有相同的相关串结构及相关串末端。由引理 11 知,它们分别具有相同的发展趋势,由引理 9,得到 $N_1(k,00) = N_1(k,01)$, $N_1(k,10) = N_1(k,11)$, 同时得到 $N_2(k,10)$ 与 $N_2(k,01)$ 的下界为 0。

下面分析 $N_2(k,00)$ 和 $N_2(k,11)$ 。由引理 9 的推论及引理 11 知只需考虑含 $(\bar{0} \sim \bar{0})$ 比特串最多的且以“01”收尾的相关串,即考虑“00”与“01”的相关串,由引理 12 知它们随着 k 的每一次增长,都要增加一个 $(\bar{0} \sim \bar{0})$ 串,且 $k = 1$, 有一个 $(\bar{0} \sim \bar{0})$ 串。那么为 k 时,有 k 个 $(\bar{0} \sim \bar{0})$ 串。由引理 9 的推论即知,它们大于 n 的相关串的个数的上界为 $O(n^{k-1})$, 也即 $N_2(k,00)$ 与 $N_2(k,11)$ 之和的上界为 $O(n^{k-1})$, 所以 $2^{-(n-1)}(N_2(k,00) + N_2(k,11)) \leq 2^{-(n-1)}n^k$, 即 $\lim_{n \rightarrow \infty} A(k) = 0$ 。

(上接第 95 页)

在该模型中语义信息库的构建是非常重要的问题,处理好语义映射机制,构建合理的语义转换规则集,是下一阶段要解决的问题。

参考文献:

[1] 史美林,向 勇. 信息化进程中的新领域——计算机协同工作技术[J]. 数据通信,1998(3):1-4.

[2] 罗海滨,范玉顺,吴 澄. workflow 技术综述[J]. 软件学报,2000,11(7):899-907.

[3] 赵卫东,戴伟辉. 基于角色的跨组织 workflow 研究[J]. 系统工程与电子技术,2003,25(8):954-958.

4 结束语

通过采用不同的比特串来分析其出现次数奇偶性的方法,证明广义自缩序列 $b(a_{k+1} + a_{k+2})$ 最小周期在所有 1024 种情形下全部达到最大,并同时证明了其具有良好的低阶自相关性。

通过大量实验,发现绝大多数广义自缩序列都存在若干比特串,能说明在大多数情形下其最小周期达到最大。因而我们猜想,对绝大多数广义自缩序列,都可以采用选择适当的多个比特串来分析其出现次数奇偶性的方法,证明其最小周期在所有情形下达到最大,即 2^{n-1} 。

一般广义自缩序列可以采用哪些比特串来完全证明其最小周期达到最大,还有待进一步研究。

参考文献:

[1] Meier W, Stafflebach O. The self-shrinking generator[C]// Advances in Cryptology - EUROCRYPT'94. Berlin, Germany:Springer-Verlag, 1995.

[2] 胡子濮,肖国镇. 关于 m 序列的自旋转序列[J]. 电子科学学报,1997,19(6):847-851.

[3] Simon R B. The linear complexity of the self-shrinking generator[J]. IEEE Trans on Inform Theory, 1999, 45(6):2073-2077.

[4] 胡子濮,张玉清. 新一类广义自缩序列的最小周期[J]. 通信学报,2003,24(6):169-176.

[5] 胡子濮,肖国镇. 第四类广义自缩序列的伪随机性[J]. 中国科学(E辑),2003,33(10):896-905.

[6] 胡子濮,张玉清,肖国镇. 对称密码学[M]. 北京:机械工业出版社,2002.

[7] 董丽华,高军涛,胡子濮. 一类广义自缩序列的伪随机性[J]. 西安电子科技大学学报,2004,31(3):394-398.

[8] Hu Yupu, Xiao Guozhen. Generalized self-shrinking generator[J]. IEEE Trans on Inform Theory, 2004, 50(4): 714-719.

[4] 王 勇. 一种基于服务的跨组织 workflow 管理系统的研究与实现[J]. 计算机工程与应用,2004(29):90-93.

[5] 常 弘,茹 锋,薛钧义. IEC61850 标准语义信息模型的互操作性分析及系统实现[J]. 电工电能新技术,2005, 24(3):58-62.

[6] 李宗华. 基于语义映射的空间数据转换及其应用研究[J]. 地理空间信息,2007,5(2):1-3.

[7] 高晓黎,盛业华,威海峰,等. 语义映射机制下 GML 与 DXF 转换方案的设计与构建[J]. 计算机应用研究,2006(4): 152-154.

[8] 万常选,郭艳阳. 电子政务语义互操作初探[J]. 电子政务, 2006(9):90-93.