

基于答案集的 Web 服务组合验证

钱俊彦^{1,2} 黄国旺² 赵岭忠²

(国防科学技术大学并行与分布处理国家重点实验室 长沙 410073)¹

(桂林电子科技大学计算机科学与工程学院 桂林 541004)²

摘 要 语义 Web 服务组合的形式化描述和验证,是保证组合服务能正确运行的重要前提基础。首先描述基于答案集编程(Answer Set Programming)的 OWL-S 建模方法,并分析基于答案集编程建模的优势。然后给出 OWL-S 流程模型中几种控制结构到中间模型 Petri 网的映射,并提出由 Petri 网生成答案集编程的算法。同时将时态约束引入到组合服务验证中,利用时态约束表达待验证性质,将验证问题转换为求解逻辑程序的答案集。最后通过一个具体的实例说明该方法的有效性。

关键词 答案集编程, Petri 网, OWL-S, 验证

中图分类号 TP311 **文献标识码** A

Answer Set Programming Based Verification of Semantic Web Service Composition

QIAN Jun-yan^{1,2} HUANG Guo-wang² ZHAO Ling-zhong²

(National Laboratory for Parallel and Distributed Processing, National University of Defense Technology, Changsha 410073, China)¹

(Department of Computer Science and Engineering, Guilin University of Electronic Technology, Guilin 541004, China)²

Abstract Formal description and verification of semantic Web service composition are the premise of the correctness of running composite services. The paper described a method to modeling OWL-S (Semantic Web service description language) based on answer set programming and analysed the advantages of this method, and the mapping of several kinds of basic control constructs in process model of OWL-S to Petri net which is the mid-mode was provided and an algorithm for generating the answer set programming was proposed. Meanwhile, introduced temporal constraints to the composite service verification to represent the property to be checked. Finally, a specific instance of the modeling and verification was applied into a case.

Keywords Answer set programming, Petri-net, OWL-S, Verification

1 引言

语义 Web 的出现给 Web 服务带来了新的契机,两者结合使机器可理解和处理 Web 服务,从而使 Web 服务的自动发现、组合、运行和监督成为可能。简单服务由于功能受限,往往不能满足实际的复杂需求。通常,复杂的需求会涉及很多来自于不同组织或者机构发布的简单服务,然而使用者很难掌握每个简单服务。Web 服务组合通过设计已有简单服务间的交互关系,协调多个服务工作,构造一个新的可提供更多功能的增值服务,满足复杂的需求。服务组合提高了 Web 服务的重要性和扩展性,降低了耦合性。由于业务需求的不断增加,且服务组合的流程本身具有复杂性和并发性,Web 服务组合日趋复杂,故保证组合方案的正确性显得格外重要。错误的组合服务不但无法完成预定任务,而且造成资源和时间的巨大浪费,因此对服务组合方案进行正确性验证有重要

的实际意义。通过对 Web 服务形式化建模、分析和验证模型来检验服务组合的正确性,是一种行之有效的办法。

国内外研究人员基于 BPELWS^[1] 和 OWL-S^[2] 对 Web 服务形式化建模、形式化验证做了大量的研究。F. V. Breugel 等对 BPEL 相关的建模方法、工具、验证等做了详细的综述^[3]。文献[4]将 BPEL 业务流程转化为 Petri 网模型来验证服务模型的安全性、死锁、循环等。文献[5]基于模型检测技术,采用工具 NuSMV 对形式化后的 BPEL 流程进行有效性及正确性验证。文献[6]提出一个框架把 BPEL 转换成 Promela 语言,然后通过工具 SPIN 验证线性时态逻辑 LTL。文献[7]通过 SPIN 对 OWL-S 流程进行检验。

Petri 网由于具有直观性、处理动态系统的能力,同时可利用数学方法对性质进行分析等优点,经常被用作建模工具。然而传统的方法主要基于可达和死锁发现,但其计算复杂度却是 P-空间完全的。随着状态的增加, Petri 网的可达集以指

到稿日期:2011-01-25 返修日期:2011-05-11 本文受国家自然科学基金(61063002, 60803033), 中国博士后基金(20090450211), 广西自然科学基金(2011GXNSFA018164, 2011GXNSFA018166), 广西研究生创新基金(2009105950812M22), 广西高等学校优秀人才资助计划等资助。
钱俊彦(1973-), 博士, 教授, 主要研究方向为软件工程、模型检验和程序验证、Web 服务; 黄国旺(1984-), 男, 硕士生, 主要研究方向为答案集、智能推理、语义 Web; 赵岭忠(1977-), 男, 博士, 副教授, 主要研究方向为形式化技术与软件验证。

数形式增长,虽然模型可能有界,但是效率非常低,不仅手工无法实现,甚至计算机都难以处理。

为此,本文提出一种新的 Web 服务验证方法,即使用答案集编程(answer set programming)^[6]对 OWL-S 流程模型进行建模。考虑到 Petri 网的图形直观性和强大的表达能力,借助 Petri 网作为中间模型。本文通过将 Petri 网模型转换到答案集编程,借助答案集编程表达能力强、智能化程度高、描述条理清晰、脉络分明、具有良好的层次性和独立性等优点来解决不完全信息下的 Web 服务验证。最重要的是答案集编程很好地弥补了 Petri 网不适合描述复杂问题的弱点。最后通过时态约束描述待验证性质,将服务验证问题转换为求解对应逻辑程序的答案集。

2 答案集编程

答案集编程是一种声明式程序,作为一个前向演绎系统,同时具备扩展析取逻辑程序和 Datalog 的特征,有效地实现了非单调推理。答案集编程经过多次扩展,当前已经可以支持扩展析取逻辑程序(Extended Disjunctive Logic Program)。一个扩展析取逻辑程序由 3 种基本元素构成:项、原子、文字。

- 项:变量或者常量;
- 原子:形如 $f(a)$ 的函数,其中 f 是一个 n 元谓词;
- 正规文字:原子或者其否定形式,即 $a, \neg a$ (\neg 为经典否定),此时称它们是互补的;
- 扩展文字:形如 $not f(a)$ 的文字,其中 not 为否定即失败(negation as failure)。

正规文字和扩展文字统称为文字。

给定任意文字集合 β ,定义正规文字 $\beta^+ = \{l | l \in \beta, l \text{ 是正规文字}\}$,负文字 $\beta^- = \{l | not l \in \beta, l \text{ 正规是文字}\}$,假设 $\beta = \{a, not \neg b, not c\}$,则 $\beta^+ = \{a\}, \beta^- = \{\neg b, c\}$ 。

扩展析取逻辑程序 P 是由规则 r 构成的集合。 r 定义如下:

$$h_1 | \dots | h_k \leftarrow b_1, \dots, b_m, not b_{m+1}, \dots, not b_n \quad (1)$$

式中, $n \geq m, b_i (1 \leq i \leq n)$ 是文字。当箭头左边头部 $head(r)$ 为空时,称规则 r 为约束,当右边主体部分 $body(r)$ 为空时,称规则 r 为事实。下面介绍一种特殊的约束——基数约束(cardinality constraint),形如:

$$x \{b_1, \dots, b_m, not b_{m+1}, \dots, not b_n\} y \quad (2)$$

式中, x, y 都是正整数,表示基数约束的上、下界。 x, y 为空时,分别默认为 0 和无穷大。式(2)中成立的文字数目在 x 和 y 之间。

逻辑程序 P 的答案集通过如下步骤获得:

首先假设 P 不包含否定即失败 not , 程序 P 中的正规文字记为 Lit , 则程序 P 的一个答案集是集合 Lit 的任意子集 S , 满足以下条件:

- (1) 对于任意规则 $h_1 | \dots | h_k \leftarrow b_1, \dots, b_m$, 如果 $b_1, \dots, b_m \in S$, 那么 $h_i \in S, 1 \leq i$;
- (2) 如果 S 包含互补的文字, 那么 $S = Lit$ 。

最后,取消上面的假设,将程序 P 关于集合 S 的约简记为 Π' , 其计算过程如下:如果规则 r 的 $body^-(r)$ 不成立, 则从程序 P 中删除规则 r , 否则把 $body^-(r)$ 从规则 r 中删除。

性质 1 如果集合 S 是约简 Π' 的答案集, 那么 S 同时也是程序 P 的答案集。

3 OWL-S 流程模型的 Petri 网模型

OWL-S^[9] 作为一种确定的、无歧义的、计算机可以理解的标记语言, 不仅描述了 Web 服务的属性和功能, 而且是连接 Web 服务和语义 Web 的桥梁。OWL-S 流程模型采用工作流的方式描述服务的流程模型, 工作流依靠数据流和控制流实现相互关联。OWL-S 流程模型通过复合流程控制其子流程的执行顺序。鉴于 Web 服务交互协议对 Web 服务的实现、应用与组合的重要意义, 应验证组合服务的控制流是否满足预期的属性。本文暂时忽略数据部分, 只抽取控制流信息转化为 Petri 网模型; 并且将子服务看成是简单服务, 下层对上层是不可见的。

图 1 分别给出 OWL-S 复合流程中的各种控制结构对应的 Petri 网模型, 图中圆圈表示子服务(子服务可能还有子服务), 黑色矩形表示迁移。

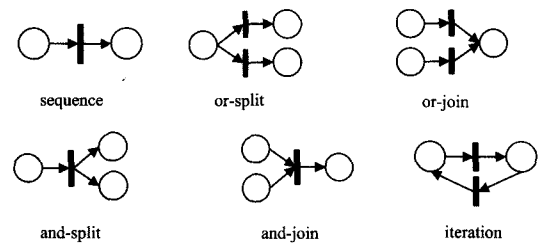


图 1 复合流程的 Petri 网表示

OWL-S 是用 XML 来编写的, 各种控制结构通过相应的标签来表示, 子结构则是采用嵌套的方法来实现, 因此整个流程模型可映射为树形结构。为了方便构造工作流, Petri 网作为过渡模型, 替换复合服务中的子服务, 直到所有的子服务都是原子服务为止。替换过程如图 2 所示。

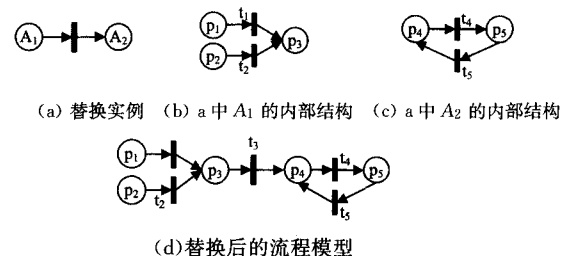


图 2 替换过程

4 Petri 网的答案集编程表示

通过上述替换过程, OWL-S 流程模型可转换为常规的 Petri 网。下面将描述 Petri 网到答案编程的转换算法。

定义 1 对于一个 Petri 网结构 $\langle P, T, F \rangle$, 其含义如下:

- (1) $P = \{p_1, p_2, \dots, p_n\}$ 是有限库所集合;
- (2) $T = \{t_1, t_2, \dots, t_n\}$ 是有限迁移集合 ($P \cup T \neq \emptyset, P \cap T = \emptyset$);
- (3) $F \subseteq (P \times T) \cup (T \times P)$ 为流关系;
- (4) 前集 $\cdot x = \{y \in P \cup T \mid F(y, x) = 1\}$, 后集 $x \cdot = \{y \in P \cup T \mid F(x, y) = 1\}$ 。

给定一个 Petri 网结构 $N = \langle P, T, F \rangle$ 和界限 n , 构建对应的逻辑程序 $\Pi_N(N, n)$ 来表示 Petri 网 N 所有可能的迁移, 但是所有的迁移都限制在 n 步范围之内。 $\Pi_N(N, n)$ 的构建步骤如下:

(1) 对于每个源库所 $p_w \in P$, 把下面规则加入到程序 $\Pi_N(N, n)$ 中:

$$\text{holds}(p_w, 0) \leftarrow \quad (3)$$

(2) 对于每个迁移 $t_w \in T$, 把下面规则加入到程序 $\Pi_N(N, n)$ 中:

$$t(w, i) \leftarrow \text{holds}(p_1, i), \dots, \text{holds}(p_n, i) \quad (4)$$

式中, $p(1), \dots, p(n)$ 是 t_w 的前集 $x, I=0, 1, \dots, n-1$ 。式(4)表示, 在第 i 步, 程序 $\Pi_N(N, n)$ 的答案集包含一个迁移实例, 当且仅当该迁移的前集 x 在第 i 步成立。

(3) 对于每个库所 $p_w \in P$, 把下面规则加入到程序 $\Pi_N(N, n)$ 中:

$$\text{holds}(p_w, i+1) \leftarrow m\{t(1, i), \dots, t(n, i)\}n \quad (5)$$

式中, $t(1), \dots, t(n)$ 是 p_w 的前集 $x, I=0, 1, \dots, n-1$ 。该规则表示, 如果 p_w 要在第 $i+1$ 步成立, 那么它的迁移前集 x 中至少有 k 个元素在第 i 步成立, 其中 $m \leq k \leq n$ 。

(4) 每个库所 $p_w \in P$, 把下面规则加入到程序 $\Pi_N(N, n)$ 中:

$$\text{holds}(p_w, i+1) \leftarrow \text{holds}(p_w, i), \text{not } t(1, i), \dots, \text{not } t(n, i) \quad (6)$$

式中, t_1, \dots, t_n 是以 p 为前集 x 的迁移集合。这是库所 p 的框架公理, 如果没有关于 p 的迁移发生, 那么 p 保持不变。

例如: 对于图 2 中的 Petri 网, 通过上述算法得到的 $\Pi_N(N, n)$ 如下:

$$\begin{aligned} &\text{holds}(p_1, 0) \leftarrow \\ &\text{holds}(p_2, 0) \leftarrow \\ &t(1, i) \leftarrow \text{holds}(p_1, i) \\ &t(2, i) \leftarrow \text{holds}(p_2, i) \\ &t(3, i) \leftarrow \text{holds}(p_3, i) \\ &t(4, i) \leftarrow \text{holds}(p_4, i) \\ &t(5, i) \leftarrow \text{holds}(p_5, i) \\ &\text{holds}(p_1, i+1) \leftarrow \text{holds}(p_1, i), \text{not } t(1, i) \\ &\text{holds}(p_2, i+1) \leftarrow \text{holds}(p_2, i), \text{not } t(2, i) \\ &\text{holds}(p_3, i+1) \leftarrow \text{holds}(p_3, i), \text{not } t(3, i) \\ &\text{holds}(p_4, i+1) \leftarrow \text{holds}(p_4, i), \text{not } t(4, i) \\ &\text{holds}(p_5, i+1) \leftarrow \text{holds}(p_5, i), \text{not } t(5, i) \\ &\text{holds}(p_3, i+1) \leftarrow t(1, i), t(2, i) \\ &\text{holds}(p_4, i+1) \leftarrow t(3, i) \\ &\text{holds}(p_5, i+1) \leftarrow t(4, i) \\ &\text{holds}(p_4, i+1) \leftarrow t(5, i) \end{aligned}$$

该例表明, 只要获得每个库所和迁移的前集与后集, 那么通过上述算法就可以获得 Petri 网的答案集编程表示。

5 扩展答案集编程

V. Lifschitz^[10,11] 将域无关的规划语言(动作语言 A, K 等)转换成答案集编程, 通过求解答案集来获得规划方案。T. C. Son 等^[12] 将时态、过程控制和分层任务网(HTN)等控制知识引入到答案集规划中, 以简化求解过程^[13]。由于服务验证只涉及服务组合的流程, 因此本文采用 Petri 网进行形式化建模并转换为答案集编程, 最后借鉴文献^[12] 的部分结论来表述 Web 服务待验证性质。

时态控制知识以时态约束的形式在答案集编程中得以实现。时态约束使用 *always*, *until*, *next* 和 *eventually* 等时态连

接词以及命题连接词 *or*, *and* 和 *negation* 来构造。本文中提到形如 $\Delta(a, b)$ 的公式, 其中 Δ 代表函数符号或者命题连接词。如果一个公式不包括自由变量, 那么该公式是封闭公式。时态约束则是在封闭公式的基础上定义的, 具体如下。

定义 2 时态约束的语法可以描述为:

- 每个封闭公式都是时态约束;
- 如果 $\omega, \bar{\omega}$ 是时态约束, 那么 *and*($\omega, \bar{\omega}$), *or*($\omega, \bar{\omega}$), *next*(ω), *until*($\omega, \bar{\omega}$), *always*(ω), *eventually*(ω) 也是时态约束。

给定状态序列 $I = \langle s_0, s_1, \dots, s_n, \dots \rangle$, f_1 和 f_2 是时态约束, f_3 是流公式(fluent formula), 以 s_i 开始的状态序列记为 I_i, t 是非负整数。下面定义时态约束成立的条件。

定义 3 I 满足 f (时态约束或者流公式), 即 $I \models f$, 当且仅当 $I_0 \models f$, 其中

- $I_i \models f_3$ iff $s_i \models f_3$
- $I_i \models \text{next}(f_1)$ iff $I_{i+1} \models f_1$
- $I_i \models \text{until}(f_1, f_2)$ iff 存在 $t \leq t_2$ 使得 $I_{i+t} \models f_2$ 并且所有满足 $t \leq t_1 \leq t_2$ 的 t_1 有 $I_{i+t_1} \models f_1$
- $I_i \models \text{always}(f_1)$ iff 所有的 $t \leq t_1$ 有 $I_{i+t_1} \models f_1$
- $I_i \models \text{eventually}(f_1)$ iff 存在 $t \leq t_1$ 使得 $I_{i+t_1} \models f_1$

检测一个时态约束对于给定的状态序列 $\langle s_T, s_{T+1}, \dots, s_n \rangle$ 是否成立, 需要定义一个特殊的谓词 $hf(F, T)$, 通过该谓词的真值来判断公式 F 对于给定的状态序列是否成立。其中 s_T 是时刻 T 的状态集合。当前的答案集求解器都要求任何在规则主体部分出现的变量都必须是域变量, 也就是说该变量必须在其他规则的头部出现。比如:

$$hf(\text{and}(F_1, F_2), T) \leftarrow hf(F_1, T), hf(F_2, T) \quad (7)$$

对于该规则必须指明 F_1 和 F_2 的域。通常的方法是给复合的时态约束重新命名, 并且定义一个新的类型——*formula* 来检测时态约束的真值, 而原子公式则不变。

- 原子公式: $\text{formula}(L) \leftarrow \text{literal}(L)$;
- 非原子公式: 对于每个时态约束 ω , 用规则集合 $r(\omega)$ 代替。

比如: 假设 $\bar{\omega} = \text{next}(\omega)$ 则 $r(\bar{\omega}) = r(\omega) \cup \{\text{next}(n_\omega, n_\omega) \leftarrow\}$ 。同样地, 若 $\bar{\omega} = \text{until}(\omega, \omega)$, 则 $r(\bar{\omega}) = r(\omega) \cup r(\omega) \cup \{\text{next}(n_\omega, n_\omega, n_\omega) \leftarrow\}$ 。

定理 1 令 C 是时态约束的集合, $I = \langle s_0, s_1, \dots, s_n, \dots \rangle$ 是一个状态序列。

$$\Pi = R_1 \cup R_2 \cup r(C) \cup r(I)$$

其中,

- (1) R_1 为普通规则以及包含命题连接词的规则;
- (2) R_2 为包含时态约束的规则;
- (3) $r(I) = \bigcup_{i=0}^n \{\text{holds}(f, t) \mid f \text{ 是一个流公式并且 } f \in s_t\}$;
- (4) $r(C) = \bigcup_{\omega \in C} r(\omega)$ 。

并可得出如下结论:

- (1) 程序 Π 有唯一的答案集 A ;
- (2) 对于 C 中的每个时态约束, $I_i \models \omega$, 当且仅当 $hf(n_\omega, i) \in A$ 。

通过定理 1, 可将时态约束的可满足性检测问题成功转换为求解相应逻辑程序的答案集。下面以货运服务为例, 其

OWL-S 到 Petri 网的转换不再详细阐述,此处直接给出中间模型,如图 3 所示,图中服务清单如图 4 所示。假设根据用户的需求得到如下待验证性质,并使用时态约束来表示:

$$holds(s_0, 0) \rightarrow eventually(holds(s_8, T)) \quad (8)$$

该性质表示,只要客户的发货请求被接受,总能得到一个明确的答复,不管是否承接请求业务。

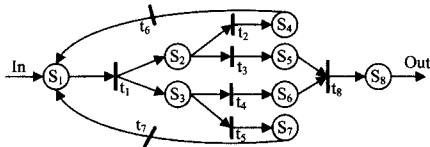


图 3 货运服务的 Petri 网模型

- S₁ 处理货运信息
- S₂ 制定档期理
- S₃ 计算价格
- S₄ 档期满
- S₅ 发送档期安排
- S₆ 发送报价
- S₇ 付款支付失败
- S₈ 发送发货信息给用户

图 4 服务清单

利用上面的方法,式(8)可以分解为如下子公式:

$$n_1(T) = eventually(holds(s_8, T))$$

$$n_2(T) = n_holds(s_0, T) \vee eventually(holds(s_8, T))$$

$n_1(T)$ 有如下规则:

$$eventually(n_1(T), holds(s_8, T)) \leftarrow time(T)$$

$n_2(T)$ 除了包含构成 $n_1(T)$ 的规则外还包含如下规则:

$$disj(n_2(T)) \leftarrow time(T)$$

$$in(n_1(T), n_2(T)) \leftarrow time(T)$$

$$in(n_holds(s_0, T), n_2(T)) \leftarrow time(T)$$

其中,谓词 $in(a, b)$ 表示 $a \in b$ 。下面提供计算时态约束的辅助公式,利用前面定义的谓词 $hf(F, T)$,向程序增加如下规则完成时态约束到扩展吸取逻辑程序的编码:

$$hf(N, T) \leftarrow always(N, N_1), hf_during(N_1, T, n+1)$$

$$hf(N, T) \leftarrow eventually(N, N_1), hf(N_1, T'), T \leq T'$$

...

$$n_hf_during(N_1, T, T') \leftarrow not\ hf(N, T''), T \leq T'' \leq T'$$

$$hf_during(N_1, T, T') \leftarrow hf(N, T), not\ n_hf_during(N_1, T, T')$$

至此,待检验性质转换完毕。

一个 OWL-S 流程模型的验证问题包括两个步骤:(1) 通过流程模型的答案集表示。运行求解机得到答案集,即流程模型的状态序列集合 $\{I_1 \dots I_i \dots\}$; (2) 对每个 I_i 构建 Π_i , 通过求解答案集检测待验证性质是否被状态序列所满足。

结合第 4 节中的算法对图 3 所示的模型进行答案集编程表示,运行求解机 clingo, 指定搜索 3 步 ($length=2$, 从 0 开始), 结果如图 5 所示。分别用每个答案集建立 I_i , 然后和上面的待检测性质分别保存为文件 check1.LP...check4.LP, 依次运行文件。除了图 6 中 Answer3 的答案集中包括 $hf(n_2, 3)$, 即式(8)成立, 其他 3 个状态序列均不满足要求。经检测发现, 模型中有无限循环, 即一旦档期满或者未付款, 服务请求者得不到答复。在 s_0, s_8 之间增加一个迁移即可解决出错问

题。也就是说不管服务请求是否被接受, 请求者都应该得到答复。



图 5 流程模型的答案集求解结果



图 6 文件 check3.LP 的运行结果

结束语 Web 服务组合是 Web 服务的一个重要研究方向, 而对 Web 服务进行形式化建模和验证是保证组合服务正确性和可靠性的前提。本文首先将 OWL-S 流程模型的基本控制结构映射为 Petri 网, 以 Petri 网为中间过渡模型, 提出 Petri 网到答案集编程的转换算法。借助答案集编程强大的表达能力、高智能化的推理和良好的独立性, 通过时态约束表达待验证性质, 将组合服务验证转换为求解扩展析取逻辑程序的答案集。本文主要侧重于静态服务组合验证。动态服务组合验证将在今后的研究中进行考虑, 该方法也有一定局限性, 其着重支持 OWL-S 控制流的验证, 对数据流和资源(包括人力资源和设备资源)支持不足。在 Petri 的库中增加 token 或在 Petri 网的弧上增加 I/O 标注来描述数据, 可支持数据流的验证, 这将在下一阶段重点研究。

参考文献

- [1] Business process executive language for Web service version 1.0 [EB/QL]. <http://www2106.ibm.com/developerworks/library/ws2bpel>, 2002
- [2] W3C Member. OWL2S: semantic markup for Web services [EB/QL]. <http://www.w3.org/Submission/OWL2S/>, December 2004
- [3] Breugel F V, Koshkina M. ModelsVerification of BPEL [EB/QL]. <http://www.cse.yorku.ca/~franck/research/drafts/tutorial.pdf>, September 2006
- [4] 王艳春, 林广艳. 基于 BPEL4WS 和 Petri 网的服务建模与分析 [J]. 系统仿真学报, 2005, (z1): 93-95
- [5] Mongiello M, Castelluccia D. Modelling and verification of BPEL business processes [C] // Proceedings of the Fourth Workshop on Model-based Development of Computer-based Systems and Third International Workshop on Model-based Methodologies for Pervasive and Embedded Software (MBD-MOMPES' 06), 2006
- [6] Kovács M, Gönczy L. Simulation and Formal Analysis of Workflow Models [C] // Proceedings of the 5th International Workshop on Graph Transformation and Visual Modeling Techniques, Vienna, Austria, 2006; 215-224
- [7] Ankolekar A, Paolucci M, Sycara K. Spinning the OWL-S process model; toward the verification of the OWL-S process models [C] // Proc of the 3rd Semantic Web Services: Preparing to Meet the World of Business Applications, a Workshop at 3rd International Semantic Web Conference (ISWC2004), 2004

(下转第 161 页)

- [26] Ma Z M, Yan Li. A literature overview of fuzzy database models [J]. Journal of Information Science and Engineering, 2008, 24 (1): 189-202
- [27] Zadeh L A. Fuzzy sets[J]. Information and Control, 1965, 8(3): 338-353
- [28] Abiteboul S, Segoufin L, Vianu V. Representing and querying XML with incomplete information[C]// Proc. 12th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, 2001: 150-161
- [29] Hung E, Getoor L, Subrahmanian V S. PXML: A probabilistic semistructured data model and algebra[C]// Proceedings of 19th the International Conference on Data Engineering, 2003: 467-478
- [30] Nierrman A, Jagadish H V. ProTDB: Probabilistic data in XML [C]// Proceedings of the 28th International Conference on Very Large Data Bases, 2002: 646-657
- [31] Senellart P, Abiteboul S. On the complexity of managing probabilistic XML data[C]// Proceedings of the 26th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, 2007: 283-292
- [32] Van Keulen M, De Keijzer A, Alink W. A probabilistic XML approach to data integration[C]// Proceedings of the 2005 International Conference on Data Engineering, 2005: 459-470
- [33] Gaurav A, Alhajj R. Incorporating fuzziness in XML and mapping fuzzy relational data into fuzzy XML[C]// Proceedings of the 2006 ACM Symposium on Applied Computing, 2006: 456-460
- [34] Oliboni B, Pozzani G. Representing fuzzy information by using XML Schema[C]// Proceedings of the 2008 19th International Conference on Database and Expert Systems Application, 2008: 683-687
- [35] Ma Z M, Yan L. Fuzzy XML data modeling with the UML and relational data models [J]. Data & Knowledge Engineering, 2007, 63: 972-996
- [36] Yan Li, Ma Z M, Liu Jian. Fuzzy data modeling based on XML Schema[C]// Proceedings of the 2009 ACM International Symposium on Applied Computing, 2009: 1563-1567
- [37] Zadeh L A. Fuzzy sets as a basis for a theory of possibility[J]. Fuzzy Sets and Systems, 1978, 1(1): 3-28
- [38] Abiteboul S, Hull R. IFO: A Formal Semantic Database Model [J]. ACM Transactions on Database Systems, 1987, 12(4): 525-565

(上接第 130 页)

示的 Web 集成式的 B/S 3 层结构模式。

结束语 物联网能够帮助企业构建基于主动反馈机制的精细化供应链,打破流程瓶颈实现库存管理实时化,改善数据采集方式,节省人力成本,提高运营效率,从根本上改善企业的运营流程,增强企业的核心竞争力。

本文研究了物联网在零售业中的应用前景,具体分析了基于物联网技术下物流信息系统在配送中心运用的改进策略,借助 RFID 技术和 SCOR 模型为企业应用物联网技术建立精细化物流配送中心提供了可行的参考模型。

参 考 文 献

- [1] 刘普合. 对于商业自动化有关问题的认识[J]. 市场与电脑, 1998 (1)
- [2] 陈东. 基于自动识别技术的物流信息系统的研究与应用[D]. 济南: 山东师范大学
- [3] 蒋亚军, 贺平, 赵会群, 等. 基于 EPC 的物联网研究综述[J]. 广东通信技术, 2005(8)
- [4] 余雷. 基于 RFID 电子标签的物联网物流管理系统[J]. 微计算机技术, 2006(2)
- [5] 谢勇, 王红卫. 基于物联网的自动入库管理系统及其应用研究 [J]. 物流技术, 2007(4)
- [6] Supply Chain Operations Reference Model. SCOR® Version 8. 0
- [7] 陈子侠. RFID 技术的应用与现代物流[J]. 商业研究, 2003(6)
- [8] 卢莉, 谢金刚. 零售业中 RFID 技术的应用的分析和对策[J]. 计算机时代, 2005(6)
- [9] 徐丹, 王铁林. RFID 在物流配送中心中的应用[J]. 物流技术, 2005(6)
- [10] Online guide to RFID Technology products. www. RFID-101.com
- [11] The EAN UCC system. www. ean-ucc.com
- [12] 斯建勇. 配送中心设施规划及仿真研究[D]. 杭州: 浙江大学
- [13] 冯淑珍. 物流中心作业系统仿真建模及资源优化配置研究[D]. 长春: 吉林大学
- [14] RFID 及其在物流配送中的应用[D]. 武汉: 华中科技大学, 2005: 24-28
- [15] 庞娟. RFID 与仓储管理系统的应用结合[J]. 中国储运, 2004 (5): 47-48

(上接第 134 页)

- [8] Gelfond M, Lifschitz V. Classical negation in logic programs and disjunctive databases[J]. New Generation Computing, 1991, 9: 365-385
- [9] Martin D, Paolucci M, McIlraith C. Bringing semantics to Web services: The OWL-S approach[C]// First Int. Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004). San Diego, CA, 2004
- [10] Lifschitz V. Action languages, answer sets, and planning [C]// The Logic Programming Paradigm-A 25 Years Perspective. New York: Springer-Verlag, 1999: 357-373
- [11] Lifschitz V. Answer set planning[C]// Schreye D, ed. Proc. of the 16th International Conference on Logic Programming (ICLP'99). The MIT Press, 1999: 23-37
- [12] Son T C, Baral C, Tran N, et al. Domain-dependent knowledge in answer set planning[J]. ACM Transactions on Computational Logic, 2006, 7(4): 613-657
- [13] Bacchus F, kabanza F. Using temporal logics to express search control knowledge for planning[J]. Artificial Intelligence, 2000, 1(2): 123-132