动态绑定实现自动服务编排

梁 平 杨宪泽
(西南民族大学计算机学院 成都 610041)

摘 要 在面对服务架构中,业务逻辑由各种服务组件组合完成,不同服务组件完成不同业务流程的业务逻辑。面对不断变化和发展的商业环境,服务的组合性需要具有灵活性和可靠性。虽然目前已有多种方法能够优化和改善面对服务系统,但是针对服务编排自动化的应用和研究较少,这主要归因于复杂的业务逻辑和繁琐的服务编排工具。介绍一种采用一阶内涵逻辑语言(FOIL)为业务流程建立其业务逻辑的FOIL公式,通过计算FOIL公式,自动生成WS-BPEL结构以动态绑定已有的服务组件,最终完成自动服务编排。Tarski的真值理论证明这种方法具有实际应用价值。

关键词 服务编排,动态绑定,FOIL,WS-BPEL,面对服务架构
中图法分类号 TP301 **文献标识码** A

Automatic Service Orchestration by Dynamically Binding

LIANG Ping YANG Xian-ze
(School of Computer Science, Southwest University for Nationalities, Chengdu 610041, China)

Abstract In a service-oriented system, business processes are completed by services. The compositionality of services to cope with changing business world is required to be flexible and robust. To complete dynamic service composition is hardly an easy one. Though there are many ways that aim to optimize and improve attributes of a SOA system, automation is little realized due to the complicated business process and tedious service composition tools. This paper introduced a way to express business process in FOIL formula and dynamically bind atomic services by computing FOIL formula and generating a BPEL structure for execution of service composition. Tarski's Truth theory proves this process is plausible in application.

Keywords Service orchestration, Dynamic binding, FOIL, WS-BPEL, Service-oriented system

尽管在真实世界中商务的业务领域是不断拓展和复杂多变的,但是在某种商业领域中的业务流程是相对不变的。如果商务系统采用了面对服务体系架构,某个问题领域的服务组合是具有针对性且可互用性的,业务流程也相对局限于该问题领域。根据文献[4],服务的可互用性会影响面对服务体系架构系统的所有属性,如某项服务在不同服务组合中的可重用性、可靠性和可访问性等。服务可互用性显示了服务如何协调工作以完成商业目标,同时避免紧耦合的问题。这对于面对服务系统的服务编排是至关重要的。目前对服务编排的应用是针对网络服务(Web Service),采用基于XML的业务过程执行语言(WS-BPEL)[2]编排网络服务,使用的是XML数据结构来表达业务逻辑。根据业务逻辑的难易程度,生成的WS-BPEL数据结构可能会是冗长和复杂的,也不容易理解和维护。一旦相关的业务流程发生变化,可能需要改写整个实现业务逻辑的WS-BPEL结构,也就很难实现动态服务编排。

针对上面的问题,本文采用FOIL公式为业务流程建模。FOIL具备足够的语法和语义来表达真实的商务世界。根据FOIL的语法规则,一个商务问题领域中的业务逻辑被解释成为FOIL句子,根据基本业务逻辑建立的FOIL句子作为基本规则,可以转换或替换产生新的业务逻辑。这些FOIL句子成为服务组合的原型,都是有内涵的且不是无限的,非常容易编写和维护,按照句子的语义可以动态绑定已有的服务组件实现自动编排。过程如图1所示。

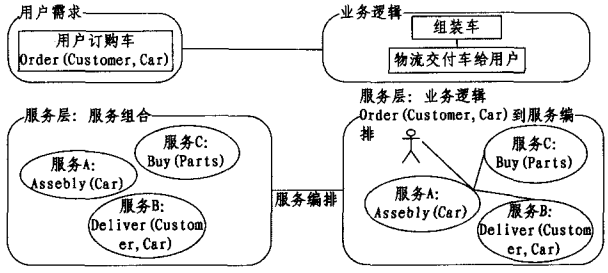

图1 相似图像I

1 业务逻辑和FOIL

根据文献[3]中的Tarski的真值理论,FOIL作为元语言描述业务逻辑,就必须材料充分和形式正确,即FOIL语句具备足够的语言机制以正确和充分地解释业务逻辑。

到稿日期:2010-07-25 返修日期:2010-11-15 本文受四川省应用基础研究项目(2008JY0070-2)资助。

梁 平(1979-),女,讲师,主要研究方向为软件工程、信息检索等,E-mail:liang_p@hotmail.com;杨宪泽(1954-),男,教授,主要研究方向为自然语言处理与数据结构等。

(1)FOIL 是模态理论语言(Modal Theory),基于 Montague 语法研究并结合了一阶逻辑语言(FOL)[5]和内涵(Intension)。FOL 语句的逻辑保证 FOIL 语句的正确性和为真值,内涵则拓展了 Montague 的内涵[1],为 FOL 加入了语义,代表了一个业务领域下的不同业务世界,在不同业务世界中的业务逻辑具有不同的语义。如用户登录验证,普通用户和高级用户在输入了正确的录信息后,进入的是不同的业务世界,提供给普通和高级用户的服务也就不同,即用户登录验证服务的语义在不同业务世界得到了不同的解释。这说明 FOIL 具有足够的语言机制表达业务逻辑。

(2)FOIL 的形式结合了 FOL 和 Montague 的内涵逻辑语法的形式,根据文献[1],已经证明是正确的。

2 业务领域建模

首先采用 FOIL 建立业务领域的模型,也就是用业务逻辑解释用户需求,业务逻辑以 FOIL 语句表达。按照下面的步骤为业务领域建模:

(1)一个业务领域虽然多变且具有拓展性,但相对于整个商业范畴有一定的业务界限。在一个业务领域中可以存在分解了的子界限,划分出这个业务领域中不同的子业务,成为业务世界。每个业务世界有独立于其他世界的界限。业务领域管理出现在这个领域中所有的变量,每个业务世界拥有一个领域变量的子集且相互排斥,即每个业务世界拥有与其他世界不同的变量子集。业务领域定义并处理一个领域层内涵的集合,对应于所有业务世界中可变的关系定义,还包括跨世界关系(当一个业务世界的关系需要其他世界中关系的支持时,业务领域能够做适当的动态绑定)。而每个业务世界具有互斥的关系集合,继承自业务领域层关系。

(2)业务领域中的业务逻辑表达为 FOIL 语句,因此业务领域层的内涵就如 CORBA 中的桩语句,根据业务逻辑涉及的变量来动态绑定业务世界中具体的逻辑关系;为业务领域建模,首先建立业务领域如下:

$M=\langle G,R,D,I\rangle$

假设 M 为业务领域;G 为 M 中业务世界的集合;D 为一个业务世界 $T(T\in G)$的域,即 $D(T)$;I 将 $D(T)$中的关系解释为业务领域 M 中的内涵 $-I(T)$,反之亦然。

1)业务领域 M 具有变量集合和关系集合以及业务世界集合 G,G 中的每个业务世界都有各自互斥的变量集合和关系集合。每个世界的变量集合为 M 变量集合的子集,每个世界的关系集合对应于 M 的内涵并具有可变的定义。

变量集 V:$x,y,\cdots$,根据 FOL 规则定义为项;

关系集 R:$P,Q,\cdots$,根据 FOL 规则定义为谓语;

业务世界集 G:$T,C,\cdots$;

内涵集 F:$f,g,\cdots$;

假设 f 为 M 中的内涵,则 $P_T(x)=f$ 指在 T 业务世界里,满足 P 关系的变量 x 具有内涵 f。

2)将用户需求解释为业务逻辑,表示为 M 的内涵 f,然后通过 $I(C)$动态绑定业务世界 $C(C\in G)$中的适当关系:

$$I(C)=\text{for all } x \text{ in } C,\langle x,P_C(x)\rangle=f(x),P\in C \text{ 且 } C\in M \quad (1)$$

如果 C 中的关系需要另一业务世界 T 中关系的支持,应在 M 中定义跨世界关系,如下所示:

$$\text{for all } x \text{ in } C,\langle x,P_C(x)\rangle=f_M(x)\rightarrow\text{for all } y \text{ in } T,\langle y,P_T(y)\rangle=f_M(y) \quad (2)$$

假设 $P_C(x),x\in C$ 和 $P_T(y),y\in T$,分别为 C 和 T 世界中的关系。

为了说明上述过程,采用一个用户订购车和用户订购零件的案例。用户订购车隐含的语义是用户希望车被生产出来并按其要求交付,而用户订购零件则暗指用户已经有车了,需要买新零件来更换已有零件。

若给定 M 的变量集 $V\{\text{Customer},\text{Car},\text{Part}_1,\text{Part}_2,\cdots,\text{Part}_n\}$,则 f_M 代表用户请求的业务逻辑内涵 Order(Customer,Car);在业务世界 C 中,对于变量集 $V_C\{\text{Customer},\text{Car}\}$和关系集 $R_C\{\text{Deliver}(\text{Customer},\text{Make}(\text{Car}))\}$,根据式(1),下列 FOIL 语句为真,

$$\text{for all Customer and Car in } C,\langle x=\text{Customer},y=\text{Car},\text{Deliver}_C(x,\text{Make}(y))\rangle=\text{Order}(\text{Customer},\text{Car}),C\in M \quad (3)$$

同样地,用户订购零件也可以这样表达,则 f_M 代表用户请求的内涵 $\text{Order}(\text{Customer},\text{Part}_1,\cdots,\text{Part}_n)$,在业务世界 T 中,对于变量集 $V_T\{\text{Customer},\text{Part}_1,\text{Part}_2,\cdots,\text{Part}_n\}$和关系集 $R_T\{\text{Deliver}_T(\text{Customer},\text{Buy}_T(\text{Part}_1,\text{Part}_2,\cdots,\text{Part}_n))\}$,根据式(1),下列 FOIL 语句为真,

$$\text{for all Customer and Part}_1,\text{Part}_2,\cdots,\text{Part}_n \text{ in } T,\langle x=\text{Customer},y_1=\text{Part}_1,y_2=\text{Part}_2,\cdots,y_n=\text{Part}_n,\text{Deliver}_T(x,\text{Buy}(y_1,y_2,\cdots,y_n))\rangle=\text{Order}(\text{Customer},\text{Part}_1,\text{Part}_2,\cdots,\text{Part}_n),C\in M \quad (4)$$

在业务世界 C 中,生产并交付车 Deliver(x,Made(x,y)需要业务世界 T 的支持,即生产车需要订购零件然后组装成车,所以在 M 中需要定义跨世界关系,

$$\text{for all Car in } C,\langle x=\text{Car},\text{Make}_C(x)\rangle==\text{for all Part}_1,\text{Part}_2,\cdots,\text{Part}_n \text{ in } T,\langle y_1=\text{Part}_1,y_2=\text{Part}_2,\cdots,y_n=\text{Part}_n,\text{Deliver}_T(\text{Buy}_T(y_1,y_2,\cdots,y_n)) \quad (5)$$

因此,给定业务领域层定义:$\text{Make}_C(x_C)=\text{Assembly}_T(\text{Buy}_T(y_T))$,$x_C$ 和 $y_T\in M$,可以得出下列语句,

$$\text{for all } x_C \text{ and } y_C \text{ in } C,\text{Deliver}(x_C,\text{Made}(y_C))\rightarrow\langle\text{for all } y_T \text{ in } T,\text{for all } x_C \text{ in } C,\text{Deliver}(x_C,\text{Make}(y_C)==(\text{Assembly}_T(\text{Buy}_T(y_T)))\rangle \quad (6)$$

上述 FOIL 语句的严格性(Rigidity)证明。如 Tarski 的真值论,

$$\text{for all } x,\text{True}(x) \text{ iff } \varphi(x),\text{且真值 true 只能出现在 } \varphi(x)\text{中} \quad (7)$$

在业务领域 M 中,由于业务逻辑为真,则 FOIL 语句为真,则给定业务领域层定义:$\text{Make}_C(x_C)=\text{Assembly}_T(\text{Buy}_T(y_T))$,$x_C$ 和 $y_T\in M$,可以得出下列语句,

$$\text{True}(\text{for all } x_C \text{ and } y_C \text{ in } C,\text{Deliver}(x_C,\text{Made}(y_C))\rightarrow\langle\text{for all } y_T \text{ in } T,\text{for all } x_C \text{ in } C,\text{Deliver}(x_C,\text{Make}(y_C)==(\text{Assembly}_T(\text{Buy}_T(y_T)))))\ \text{iff}\ (\text{for all } x_C \text{ and } y_C \text{ in } C,\text{Deliver}(x_C,\text{Made}(y_C))\rightarrow\langle\text{for all } y_T \text{ in } T,\text{for all } x_C \text{ in } C,\text{Deliver}(x_C,\text{Make}(y_C)==(\text{Assembly}_T(\text{Buy}_T(y_T)))))$$

3 映射 FOIL 语句到服务组合

服务是功能的集合,具有要完成的服务合同,而服务合同

需要一个或多个输入和输出。从输入到输出，服务具有处理请求并返回答复的机制。如网络服务包含了进程、序列、流和链接。业务逻辑的语义可以用词组结构表达，同样地 FOIL 的语句也采用词组结构作谓语，这样保证了业务逻辑语义的真值。然而 FOIL 无法表达 WS-BPEL 具有的并发、遍历和外部链接等结构。为了解决这个问题，本文引入 FOL 的命题(Proposition)符号来支持服务的并发、遍历和外部链接语义，如下所示：

for all x in T, $\langle x, P(Q(x))\rangle = f \rightarrow$ iff P 与 $Q(x)$ 链接(外部链接 link)且在 $Q(x)$ 完成后顺序完成(sequence)，f 在 T 业务世界内为真。

for all x, y in T, $\langle x, y, P(x) \wedge Q(y)\rangle = f \rightarrow$ iff $P(x)$ 和 $Q(x)$ 并发(并发 flow)，f 在 T 内为真。

for all x in T, $\langle x, (P(\forall x))\rangle = f \rightarrow$ iff 对于变量 x，$P(x)$ 都具有 f(遍历 foreach)，f 在 T 内为真。

根据上述命题，FOIL 是能够满足网络服务的要求的。

4 自动化从 FOIL 语句到服务编排

由于 FOIL 语句的材料充分和形式正确，再加入了命题符号辅助表达并发、遍历和外部链接，一段短的伪码就能实现从 FOIL 语句到基于 XML 结构编排网络服务的 WS-BPEL 结构，这样实现了动态绑定，也就实现了服务编排的自动化。

```
Read a formula;
If (P(x) ∧Q(y))    //并发和顺序
Then ⟨Sequence⟩⟨flow P⟩⟨flow Q⟩⟨/Sequence⟩
If (P(Q(x)))    //链接和顺序
Then ⟨PartnerLink P⟩⟨PartnerLink Q⟩
If (P(for all x))    //遍历
Then ⟨foreach (p(x))⟩
```

结束语 FOIL 语句作为业务逻辑的元语言，证明了是材料充分和形式正确的。引入命题符号后，FOIL 就能够表达 WS-BPEL 的结构，满足服务编排的需求。给出的伪码显示了从业务逻辑到服务编排自动化的可能性，避免了书写 WS-BPEL 的繁琐，还容易维护。动态绑定服务组件是服务编排自动化的实现。

参 考 文 献

[1] Fitting M. First Order Intensional Language[J]. Annals of Pure and Applied Logic, 2004, 127(1-3): 171-193

[2] Web Services Business Process Execution LanguageVersion 2.0, OASIS Standard[EB/OL]. http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html, 2007-04-11

[3] Hodges W. Tarski's truth definitions. In the Stanford Encyclopedia of Philosophy[EB/OL]. http://plato.stanford.edu/entries/tarski-truth/Fenstermacher, 2001

[4] Josuttis N M. SOA in Practice: The Art of Distributed System Design (Theory in Practice)(1 edition)[M]. O'Reilly Media, August 24, 2007

[5] Magnus P D. Forall x: an introduction to formal logic[M]. version 1.24 [080109], University at Albany, State University of New York (tutorial)

(上接第 164 页)

参 考 文 献

[1] Hummel O, Atkinson C. Using the Web as a Reuse Repository [C]// Proceedings of the International Conference on Software Reuse (ICSR-9). 2006: 298-311

[2] Li Yan, Liu Yao, Zhang Liang-jie, et al. An Explortary Study of Web Services on the Internet[C]// Proceedings of the International Conference on Web Services (ICWS). 2007: 380-387

[3] Fan J, Kambhampati S. A Snapshot of Public Web Services[J]. ACM SIGMOD Record, 2005, 34(1): 24-32

[4] Dekel U, Herbsleb J D. Improving API documentation usability with knowledge pushing [C]// Proceedings of IEEE 31st International Conference on Software Engineering. 2009: 320-330

[5] Seacord R C, Hissam S A, Wallnau K C. AGORA: a Search Engine for Software Components [J]. IEEE Internet Computing, 1998, 2(6): 62-70

[6] Platzer C, Dusdar S. A Vector Space Search Engine for Web Service[C]// Proceedings of the Third European Conference on Web Services (ECOWS). 2005: 62-71

[7] Kim Jinhan, Lee Sanghoon, Hwang Seung-won, et al. Towards an Intelligent Code Search Engine [C]// Proceedings of Twenty-Fourth Conference on Artificial Intelligence (AAAI-10). Atlanta, Georgia, USA, July 2010: 1358-1363

[8] Kim Jinhan, Lee Sanghoon, Hwang Seung-won, et al. Adding Examples into Java Documents[C]// Proceedings of the 24th IEEE/ACM International Conference on Automated Software Engineering (ASE 2009). Nov. 2009: 540-544

[9] http://www.google.com/codesearch

[10] http://maven.apache.org/

[11] http://www.sourceforge.net/

[12] http://www.componentsource.com/

[13] RFC 1321, R. L. Rivest. The MD5 Message Digest Algorithm [S]

[14] 郭庆琳，李艳梅，唐琦. 基于 VSM 的文本相似度计算的研究. 计算机应用研究[J]. 2008, 25(11): 3256-3258

[15] 曹恬，周立，张国煊. 一种基于词共现的文本相似度计算[J]. 计算机工程与科学, 2007, 29(3): 52-73

[16] Pang Bo, Lee Lillian. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimun Cuts [C]// Proceedings of the ACL. 2004: 271-278

[17] Hatzivassiloglou V, McKeown K R. Predicting the semantic orientation of adjectives [C]// Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics. 1997: 174-181

[18] Wiebe J, Bruce R, Bell M. Learning subjective language[J]. Computational Linguistics, 2004, 30(3): 277-308

[19] TSR[EB/OL]. http://tsr.trustie.net