

一种基于同构群的构件接口适配方法

祁华成, 陆 英

(昆山开放大学信息与自动化系, 江苏 昆山 215300)

摘 要: 构件化软件一直面临着构件组装过程的失配问题。对此, 本文利用形式化方法建立了构件接口模型, 提出了端口同构群和等价端口的概念, 并证明了交互构件之间通过等价端口实现相互通信。最后, 通过实例应用阐述了构件端口适配方法步骤, 建立适配器解决等价端口的失配问题, 为构件化软件接口之间的不匹配提供了一种可行的解决方法。

关键词: 件接口模型; 同构群; 等价端口; 适配器

中图分类号: TP3-0 **文献标识码:** A

A Method of Component Interface Adaptation Based on Isomorphic Group

QI Huacheng, LU Ying

(Information and Automation Department, Kunshan Open University, Kunshan 215300, China)

Abstract: Component-based software is always facing the problem of mismatch in assembly process. To solve this problem, component interface model is established using the formal method, the concept of port isomorphic group and equivalent port are proposed. It proves that the equivalent ports can call each other through the reconfiguration port in the paper. Through an example, the steps of component port adaptation are given, and the adapter solves the problem of the mismatch of the equivalent port, which provides a feasible solution for the mismatch between the component-based software syntax.

Keywords: component interface model; isomorphic group; equivalent port; adapter

1 引言(Introduction)

随着Web服务和网构软件^[1]概念的提出, 通过组装已有构件产生新的软件, 焦点在于复用, 复用有三方面的要求: 构件实体具有通用性、构件功能具有完备性、构件接口具有兼容性^[2]。由于这三方面相互对立, 因此, 组装第三方构件失配现象是不可避免的, 本文工作重点是自动化生产构件端口适配器解决构件接口失配问题。

为了增强构件的复用度, 国内一些学者从二进制代码重用^[3]、特定领域软件复用^[4]、模型驱动简化自适应系统^[5]出发做了相关研究, 目前, 已有大量工作研究构件适配方法并延伸到Web服务等领域, 从适配方法上分为限制性方法和生成性方法^[6], 限制性方法是指从构件协议中删除引起失配路径的方法, 缺点是构件丧失了部分功能, 可能导致不可预测的失陪问题; 生成性方法是指不更改构件行为与功能, 通过重排序消息及数据生成适配器协调的方法, 优点保留构件功能完备性, 缺点是需要修改构件接口语法, 目前的研究主要使用生成性方法。针对接口失配, 文献[7]探讨了如何在接口映射前提下, 依赖于分析有限状态机的所有执行轨迹生成适配器;

文献[8]基于PI演算描述构件接口行为, 然后给出适配器生成方法, 该方法只能解决二元适配问题, 并且计算复杂性较高; 文献[9]基于接口映射规则生成Web服务适配器, 该方法产生的适配器没有控制逻辑, 一方面不能解决复杂Web服务适配场景, 另一方面算复杂度高; 文献[10]利用面向方面的编程技术把适配方面织入到需适配的Web服务中, 对失配模式进行了分类, 阐述每种失配解决的适配思想, 该方法需要了解构件的内部结构。这些研究存在着很大的改进空间: (1) 计算复杂度高, 只是部分缓解了适配的压力; (2) 需要了解构件内部行为, 弱化了构件黑盒子原则; (3) 忽略了构件适配可以划分难度级别: 接口层、行为层、语义层, 构件接口是构件组装交互的唯一通道, 接口层适配是构件化软件的首选。

针对以上的分析, 本文利用形式化方法建立了构件接口模型, 提出了基于同构群的构件接口适配方法, 并证明了交互构件之间通过等价端口实现相互通信, 并给出了完整的适配步骤, 一方面将构件适配限制在接口层, 降低了构件适配的复杂度; 另一方面遵循构件化组装黑盒子原则, 提高了构件复用度。

2 构件接口模型(Component interface model)

构件接口特指构件之间联接的语法规则，独立于构件行为与功能，分离于构件的实现。构件接口是由一组端口构成的集合，每个端口包含端口的名称，参数名称、参数类型、返回类型、及异常类型。

定义2.1 端口(Port)。端口是一个四元组， $Port := \langle Port_name, Parameter_map, Return_type, Exception_type \rangle$ ，其中：

Port_name表示端口名称；

Parameter_map是一组参数类型和参数名组成的有序集合，记为： $\langle Parameter_type, Parameter_name \rangle$ ，其中，Parameter_type, Parameter_name并不具有唯一性即可重复，但对于成对的一组 $\langle Parameter_type, Parameter_name \rangle$ 在Port中的位置是有顺序的；

Return_type表示Port返回值类型；

Exception_type表示端口可能发生的异常类型。

根据提供消息的方向，可以将Port分为输入端口和输出端口，分别记为：Input_port和Output_port；从提供的服务范围可分为哑端口和显端口哑端口仅为构件本身提供服务，显端口是指与外界进行信息交流的端口。

定义2.2 接口(Interface)。接口是一个四元组， $Interface := \langle interface, Interface_name, Onput_port_set, Input_port_set \rangle$ ，其中：

interface是所有接口的标示符；

Interface_name表示接口的名称，在同一域中，它是具有唯一性；

Onput_port_set是输出端口构成的集合；

Input_port_set是输入端口构成的集合。

接口是构件与外界交流的唯一通道。

3 同构群及等价端口(Isomorphic group, equivalent port)

构件之间接口相互通信前提接口中的端口之间需要具有兼容性，同构群是一个集合，用其判断构件端口之间的兼容性，其定义如下：

定义3.1 同构群(Isomorphic_group)。设集合 $S = \{s_1, s_2, s_3, \dots, s_n\}$ ，以S的元素全排列为元素构成的集合称为S的同构群，记为： $Isomorphic_group_S$ 。

可得出， $S \in Isomorphic_group_S$ ， $Isomorphic_group_S$ 中的元素个数为 $|S|!$ 。例如： $S = \{1, 2, 3\}$ ，则可以推出 $Isomorphic_group_S = \{\{1, 2, 3\}, \{2, 3, 1\}, \{3, 1, 2\}, \{1, 3, 2\}, \{2, 1, 3\}, \{3, 2, 1\}\}$ 。

为了判断构件端口之间的兼容性，本文提出了端口参数集合、端口集合和端口同构群概念，其定义如下：

定义3.2 端口参数集合(Port_parameter_set)。Port_parameter_set是由端口(Port)参数类型构成的有序集合，并且Port中的参数类型的顺序与Port_parameter_set中的元素顺序一一对应。

定义3.3 端口集合(Ports)。Ports是由端口的返回值类型、参数类型，以及异常类型构成的有序集合，并且端口中的返回值类型、参数类型、异常类型的顺序与Ports中的元素顺序一一对应。

由端口集合的定义可知，Ports与集合 $\{Return_type, Port_parameter_set, Exception_type\}$ 是一一对应的。

定义3.4 端口同构群(Port_isomorphic_group)。假设Port为一个端口，Port_parameter_set为对应的端口参数集合，由Port_parameter_set对应的群的每个元素前后分别加入Return_type, Exception_type构成的一个特殊的群称为端口同构群。

例如：一个端口为`int before(boolean arg0, Object[] arg1, boolean arg2) throws IOException`。可得对应的 $Port_parameter_set = \{\text{boolean}, \text{Object}[], \text{boolean}, \}$ 与 $Port_isomorphic_group = \{\{\text{int}, \text{boolean}, \text{Object}[], \text{boolean}, \text{IOException}\}, \{\text{int}, \text{boolean}, \text{boolean}, \text{Object}[], \text{IOException}\}, \{\text{int}, \text{Object}[], \text{boolean}, \text{boolean}, \text{IOException}\}\}$ 。

由端口集合及端口同构群可知：

(1) 端口同构群是一个以端口参数集合为基础的特殊同构群；

(2) 端口同构群与端口参数集合对应的同构群的元素个数相同，为 $|Port_parameter_set|! / |\text{boolean}|! = 3! / 2! = 3$ ；

(3) 端口集合是端口同构群中的一个元素，即 $Ports \in Port_isomorphic_group$ 。

定义3.5 等价端口(Equivalent_Port)。假设两个端口(Port1, Port2)，如果Port1与Port2对应的端口集合属于同一端口同构群，称Port1与Port2之间具有等价性，具有等价性的端口称为等价端口。

根据定义3.5可知：等价端口的实质就是端口同构群元素之间的关系。

定理1 交互构件之间通过等价端口实现相互通信。

证明：假设交互构件对应的两个端口(Port1, Port2)是等价端口，根据定义3.5 $\Rightarrow Port1, Port2$ 属于同一端口同构群，由定义3.4和定义3.3 $\Rightarrow Port1, Port2$ 返回值类型、异常类型一致，且Port1端口参数集合与Port2端口参数集合的参数类型与个数具有一致性，差别在于参数顺序不同，因此，Port1, Port2仅需重构端口顺序便可实现相互通信。

定理1拓宽了构件本体复用的兼容性，将具有两项特征的构件视为同质构件，即功能相同和端口等价。

4 构件接口适配方法(Adaptation method of component interface)

构件之间交互，首先，构件检索条件是Interface中的Port返回值类型必须满足需求；其次，Port中的异常必须与需求的异常是同质的即同种类型的异常包括异常的子类型；最后，Port参数类型与顺序必须满足需求。

基于同构群的构件接口适配分为四个步骤(自动化工具完成)：

第一步：建立Input_port对应的集合(Input_Ports)，根据Input_Ports得到端口同构群(Inputport_isomorphic_group)；

第二步：建立Output_port对应的集合(Output_Ports)，根据构件维度匹配检索的算法限制，Output_Ports一定是Inputport_isomorphic_group元素，如果Input_Ports与Output_Ports对应Inputport_isomorphic_group中的同一元素，直接返回不需适配，否则，进入第三步；

第三步：以Input_port为端口，以Output_port为被调用端口建立适配器，因为Input_port与Output_port是等价端口，故适配器得到很简单的；

第四步：测试并修改构件接口(Interface)对应的规格说明。

基于驱动器的构件接口适配的实质是接口中端口的参数顺序调整的过程。

5 实例应用研究(Application research)

为了阐明本文提出的基于同构群的构件接口适配方法，我们采用构件化技术自行开发了Java Web应用程序：资源管理系统，该系统业务层由34个构件组装而成，构件集合用CS表示：

CS={CardAction, CardBean, CardDao, CardForm, Retrieve, ApplicationResources, UserAction, UserBean, Userdao, UserForm, DBPool, PageBean, Pagination, db, ...}

为了尽可能简化实例，仅以CS中的一个构件即数据持久层中的db接口为例，db接口包括四个端口(Insert、delete、update、select)增、删、改、查端口。如表1所示，构件交互输入与输出update端口。输入与输出端口update之间不能进行组装使用，因此需要对端口进行适配。

表1 端口及集合元素
Tab.1 Port and sets

端口	端口及集合元素
输入端口	public static boolean update(int Id, ArrayList list, boolean return) throws IOException
输出端口	public static boolean update(ArrayList list, int Id, boolean return) throws IOException

首先，检查输入端口和输出端口的返回值类型、异常是否同质，由表1可以得出需求端口和提供端口的返回值类型、异常是同质：public static boolean、IOException。

其次，为输入端口建立端口参数集合update_parameter_set={int, ArrayList, boolean}。端口同构群update_isomorphic_group={{boolean, int, ArrayList, boolean, IOException}, {boolean, int, Boolean, ArrayList, IOException}, {boolean, ArrayList, int, boolean, IOException}, {boolean, ArrayList, boolean, int, IOException}, {boolean, Boolean, int, ArrayList, IOException}, {boolean, boolean, ArrayList, int, IOException}}

再次，判断输入端口与输出端口是否等价端口，输入端口集合Input_update_ports={Boolean, int, ArrayList, boolean, IOException}，输出端口集合Output_update_ports={Boolean, ArrayList, int, boolean, IOException}，由于Input_update_ports与Onput_update_ports同属于同一端口同构群update_isomorphic_group。因此，Input_update_port与Onput_update_port是等价端口，根据定理1=>供、需update端口之间可以通过适配端口进行调用。

最后，建立适配器Adapter_update，实现输入与输出update端口之间通信，如图1所示。update端口适配，该适配器通过工具自动生成。

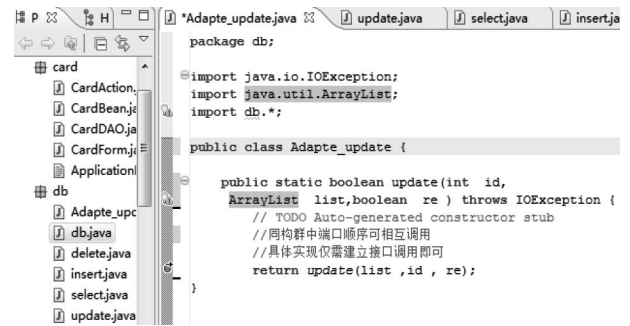


图1 update端口适配

Fig.1 Update adaption of port

6 结论(Conclusion)

本文面向构件化软件开发模式，针对构件接口之间的交互，基于同构群理论，提出了一种基于同构群的构件接口适配方法。此方法根据同构群理论给出等价端口的定义，同时证明了交互构件之间通过等价端口实现相互通信等，其适配实质是端口的参数不兼容性修改的过程，通过自动化生成适配器解决等价端口的失配问题，为构件化软件语法之间的不匹配提供了一种可行的解决方法。

参考文献(References)

- [1] 杨美清,吕建,梅宏.网构软件技术体系:一种以体系结构为中心的途径[J].中国科学E辑:信息科学,2008,38(6):2991-3000.
- [2] S.Becker, A.Brogi, I.Gorton, et al. Towards an Engineering Approach to Component Adaptation[C]. Proc of Architecting Systems with Trustworthy Components, Springer-Verlag Berlin Heidelberg, 2006:193-215.
- [3] 彭国军,梁玉,张焕国,等.软件二进制代码重用技术综述[J].软件学报,2017,28(8):2026-2045.
- [4] 于文静,赵海燕,张伟,等.基于特征模型的软件产品自动导出方法综述[J].软件学报,2016,27(1):26-44.
- [5] 赵天琪,赵海燕,张伟,等.基于模型的自适应方法综述[J].软件学报,2018,29(1):23-41.
- [6] Radu Mateescu. Adaptation of service protocols using process algebra and on-the-fly reduction technique[J]. IEEE Trans. On Software Engineering, 2012, 38(4):755-777.
- [7] Yellin DM, Strom RE. Protocol specifications and component adapters[J]. ACM Trans on Programming Languages and Systems, 1997, 19(2):292-333.
- [8] Bracciali A, Brogi A. A formal approach to component adaptation[J]. Journal of System and Software, 2005, 74(1):45-54.
- [9] Li XT, Fan YS. A pattern-based approach to protocol mediation for Web services composition[J]. Information and Software Technology, 2010, 52(3):304-323.
- [10] Kongdenfa W, N ezhda H. Mismatch patterns and adaptation aspects: a foundation for rapid development of Web service adapter[J]. IEEE Trans. on Services Computing, 2009, 2(2):94-107.

作者简介:

祁华成(1983-),男,硕士,系统架构设计师.研究领域:软件体系结构,构件技术.

陆英(1978-),女,硕士,高级讲师.研究领域:软件工程.