

基于资源库——规约 PO 查询的研究与实现

龚学海, 郭朝珍

(福州大学 数学与计算机科学学院, 福建 福州 350002)

摘要:持久化对象查询是大多数面向对象系统的基本任务之一,但传统的常规查询方法存在着可重用性差、可维护性不佳的缺陷。针对这个问题,提出了“约束条件逻辑”的新概念,并在此基础上对原始的资源库——规约方法进行改造,从而提出了一种新的持久化对象查询方法:扩展的资源库——规约方法。试验表明,该方法具有良好的可重用性与可维护性。

关键词:持久化对象;扩展的资源库——规约方法;约束条件逻辑

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2009)05-0009-04

Research and Realization of Repository - Specification Based Query Method of Persistent Objects

GONG Xue-hai, GUO Chao-zhen

(College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350002, China)

Abstract: Querying persistent objects is a fundamental task in most object-oriented systems, but the ordinary query method is considered to be lacking in reusability and maintainability. Improves the original repository-specification method base on the new conception: "where condition logic", and it presents a new query method: extended repository-specification method, which can be applied to persistent objects. And experiments show that the method has much higher reusability and maintainability.

Key words: persistent object; extended repository-specification method; where condition logic

0 引言

通常,持久化对象(PO)特指运行状态信息保存在关系型数据库的领域对象,如何对持久化对象进行有效查询,历来都是研究热点之一。

常规查询方法(简称常规方法)为每个持久化对象建立特定的数据访问对象,每个数据访问对象包含若干查询函数,每个查询函数仅针对一种约束条件进行查询^[1]。常规方法虽然直观,但随着查询约束条件数目与复杂度的增加,数据访问对象规模迅速增大,严重降低了软件的开发和维护效率。

Martin Fowler 与 Eric Evans 共同提出了资源库——规约方法^[2](Repository - Specification Method)。为了便于描述,文中把这种方法称为原始的资源库——规约方法(Original Repository - Specification Method),简称 ORS 方法。该方法利用规约来筛选领

域对象,十分简洁且易于重用^[3],但方法假定待筛选的领域对象都是内存对象,因而无法直接应用于持久化对象查询。

尽管如此,ORS 方法仍具有极大的参考价值。经过改进,文中提出了一种持久化对象查询的新方法:扩展的规约——资源库方法(Extended Repository - Specification Method),简称 ERS 方法。试验证明,与常规方法相比,ERS 方法具有更好的可复用性和可维护性,效率也略高。

1 原始方法无法查询持久化对象的根源

ORS 方法不适用于查询持久化对象。其根源在于:ORS 方法假设规约自身就能解析封装的业务规则^[4]。但是,持久化对象的状态信息存储在数据库中,规约必须与数据库引擎协作,才有可能完成查询持久化对象的任务。

2 资源库与规约的持久化特性扩展

由于数据库查询引擎的参与,持久化对象查询就不可避免地使用到数据库查询模型。数据库查询模型是一种通用的查找,筛选关系数据的模型。为了扩展

收稿日期:2008-08-28

基金项目:福建省自然科学基金计划资助项目(A0710007)

作者简介:龚学海(1984-),男,硕士研究生,研究方向为 CSCW;郭朝珍,教授,硕士生导师,研究方向为 CSCW,IGDSS,分布式数据库技术。

资源库与规约,文中把数据库查询模型拆分成几个部分:

(1) 表头格式(T):表示查询结果集的数据结构,包括查询作用的数据库表以及相应的列。

(2) 约束条件(C):表示查询时的数据筛选条件。

(3) 附加条件(A):表示查询结果集的组织形式。

(4) 结果集定义(S): $S = T + A$ 。

(5) 数据库查询命令(D): $D = T + C + A$ 。

通过对持久化对象查询的研究与总结,文中发现了一条重要规律:

规律 1:

假设有以下的符号定义:

DomainObject:代表某一特定的持久化领域对象。

DomainObjectDao:代表持久化对象对应的数据库访问对象。

$\text{CriteriaSet} = \{\text{Criteria}_k \mid 1 \leq k \leq n, n \in N\}$:代表针对 DomainObject 的查询约束条件集合,其中 Criteria_k 代表第 k 个约束条件。

$\text{QueryFunctionSet} = \{\text{QueryFunction}_i \mid 1 \leq i \leq n, n \in N\}$:代表 DomainObjectDao 的查询成员函数集合。其中 QueryFunction_i 代表第 i 个查询成员函数。

对 $\forall \text{DomainObject}$,总 \exists 至少一个 DomainObject Dao 的设计方式,使得:

$\forall \text{QueryFunction}_i, \text{QueryFunction}_j \in \text{FunctionSet}$,有以下关系:

(1) QueryFunction_i 与 QueryFunction_j 的结果集定义相同。

(2) QueryFunction_i 与 QueryFunction_j 的约束条件不一定相同。

因此,结果集只需保存一份副本,通过组合不同的约束条件,就可构成各种数据库查询命令。

资源库负责处理所有与领域对象查询有关的逻辑^[4],包括记录类中成员变量与数据库表属性字段的对应关系,因此资源库是封装结果集定义的最佳场所;而规约包含业务规则,业务规则体现在约束条件中,因此规约是约束条件的最佳归宿。资源库调用特定的规约,组合约束条件与结果集定义,成为完整的数据库查询命令,进而在数据库引擎协作下完成持久化对象查询功能。

3 应用模板

ERS 方法的应用模板可以用 UML 类图与顺序图来表示,只需稍加修改,模板就可应用于实际软件开

发。图 1 展示了应用模板类图。

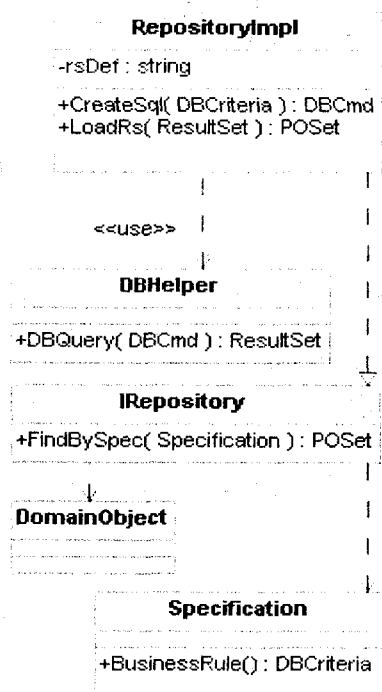


图 1 ERS 方法的模板类图

图 1 中涉及到的类有:

(1) DomainObject:领域对象类。

(2) Specification:规约类。

(3) RepositoryImpl:资源库的具体实现类。

(4) IRepository:资源库接口类。

(5) DBHelper:数据库帮助类,用于封装具体数据库查询的相关逻辑。

各成员函数及成员变量的含义如下:

(1) BusinessRule():返回规约封装的约束条件 DBCriteria。

(2) FindBySpec(Specification):资源库内唯一的查询函数。函数接受一个特定的规约,返回持久化对象结果集 POSet。

(3) rsDef:代表领域对象对应的结果集定义。

(4) CreateSql(DBCriteria):函数将结果集定义与规约提供的约束条件 DBCriteria 拼接,返回完整的数据库查询命令。

(5) DBQuery(DBCmd):函数接受一个数据库查询命令 DBCmd,通过内部查询,返回得到的关系数据结果集 ResultSet。

(6) LoadRs(ResultSet):函数根据关系数据结果集组装相应的持久化对象,并返回持久化对象结果集 POSet。

ERS 方法通过上述类的相互协作,完成持久化对象查询工作。具体交互过程如图 2 所示。

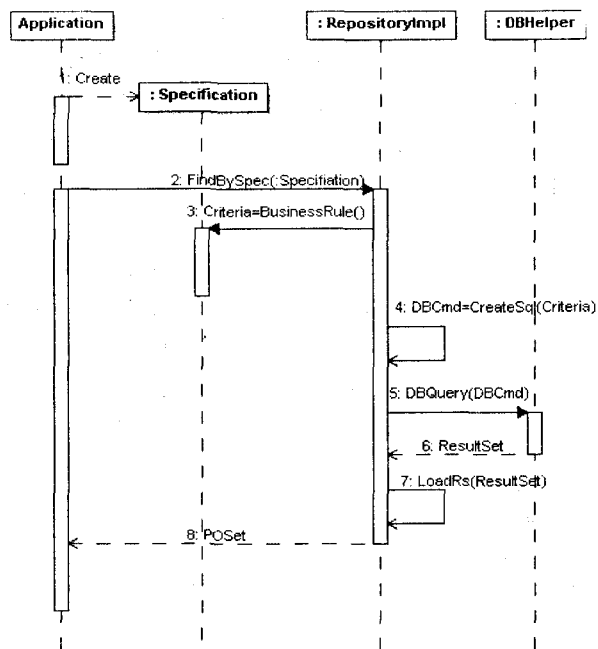


图2 ERS方法的查询步骤

4 复合规约

4.1 约束条件逻辑

ERS方法使用复合规约来处理复合查询任务。

复合规约就是简单规约的组合。由于规约封装了约束条件,所以规约组合的实质就是约束条件的组合。

文中发现:约束条件组合有着类似于布尔代数的基本规律,文中提出了一套描述约束条件组合的符号集以及若干基本运算律,它们构成了约束条件逻辑(Where Condition Logic)。

定义1:约束条件逻辑是一个集合 B ,它有两个二元运算符 \vee, \wedge , 一个一元运算符 $!$, 以及一个字符串形式的约束条件集合 S , 并有下列性质成立:

对 $\forall x, y \in S, \exists z \in S$

$x \wedge y = x \text{ And } y = z \in S$

$x \vee y = x \text{ Or } y = z \in S$

$!x = \text{Not } x = z \in S$

称 $x \wedge y$ 为约束条件与运算, $x \vee y$ 为约束条件或运算, $!x$ 为约束条件非运算, 三种运算统称约束条件运算。其中, 约束条件与运算, 约束条件或运算统称约束条件双目运算, 约束条件非运算又称为约束条件单目运算。符号 $\wedge, \vee, !$ 分别为对应的逻辑运算。

4.2 基于约束条件逻辑的规约继承体系

利用约束条件逻辑的相关理论可以扩展ERS方法中的规约模型, 建立的适合复合查询的规约继承体系如图3所示。

图4显示了双目规约 BinarySpec 类的详细结构: 成员变量 specA, specB, binaryOper 依次代表约束条件

的操作数 A, 操作数 B 和双目运算符。为了复合规约, BinarySpec 类重载了 BusinessRule() 方法, 伪代码描述为:

BinarySpec.Specification

* 重载 BusinessRule(): string 成员函数

override DBCriteria BusinessRule()

DBCriteria result = null;

/* 将 specA 的约束条件、双目运算符、specB 的约束条件进行与运算(在伪代码中用“&&”表示约束条件与运算符) */

result = specA.BusinessRule() && binaryOper && specB.BusinessRule();

/* 返回组合成的约束条件。 */

return result;

其他复合规约的构造思路与之类似。

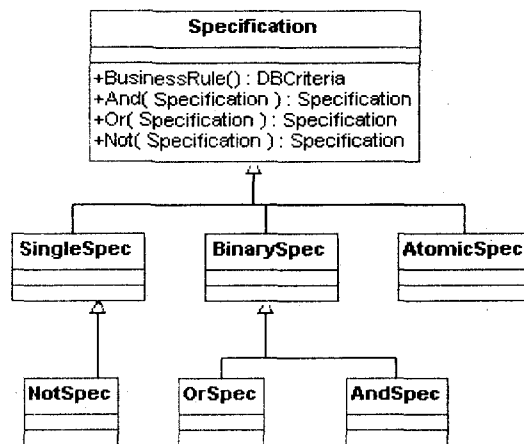


图3 规约继承体系

BinarySpec

-specA: Specification

-specB: Specification

-binaryOper: string

+BinarySpec(Specification, Specification)

图4 双目规约的类图

5 应用 ERS 方法的一个实例——Mini BBS 系统

Mini BBS 系统是一个试验性质的原型系统, 系统分别用常规方法与 ERS 方法实现持久化对象查询功能。

5.1 应用背景描述

Mini BBS 系统包含以下领域对象:

(1) Topic 类: 帖子类, 包含 Id 号, 帖子名称, 发帖时间等属性, 一个帖子属于一个特定的主题, 并有一个

特定的作者。

(2)Author类:作者类,包含 Id 号,作者姓名等属性。

(3)Subject类:主题类,包含 Id 号,主题名称等属性。

系统包含了几个查询任务:

任务 1:查找某作者发表的所有帖子。

任务 2:查找属于某个主题的所有帖子。

任务 3:查找在不早于某时间点发表的所有帖子。

任务 4:查找属于某个主题的,由某作者在不早于某时间点发表的所有帖子。

5.2 用常规方法实现查询任务

常规方法在 Topic 类的数据库访问对象中添加 4 个对应的查询成员函数。虽然可以通过优化技术重用部分代码,但数据库访问对象的规模仍然偏大。

5.3 用 ERS 方法实现查询任务

ERS 方法建立如下简单规约,完成查询任务:

(1)ByAuthorNameTopicSpec 类:用于描述特定作者名称的规约。

(2)BySubjectNameTopicSpec 类:用于描述特定主题名称的规约。

(3)AtOrLaterTopicSpec 类:用于描述不早于某个时间点的规约。

这些简单规约,分别解决任务 1,2,3,将这 3 个简单规约进行“^”运算,便可完成任务 4。

6 常规方法和 ERS 方法的综合性能对比

6.1 试验方法

采用 visual studio 2008 中相关工具计算代码矩阵(code metric),来评价常规方法和 ERS 方法的可重用性与可维护性。

采用 visual studio 2008 中的性能测试工具来比较常规方法和 ERS 方法的执行效率。具体步骤是:在消除数据库缓存影响的情况下,分别运行常规方法与 ERS 方法 10000 次,并记录消耗时间。

6.2 试验结果

表 1 记录了代码矩阵计算结果,用于测试的数据访问对象(常规方法)已进行过代码优化处理^[5~7]。

比较发现:资源库代码与优化后的数据库访问对象代码比较,在可维护性指数,继承深度与类耦合方面都达到了优秀水平。但是,资源库代码的圈复杂度,代码行数得分均比数据库访问对象代码更优^[8]。当复杂查询条件较多时,ERS 方法的资源库代码比经过优化的常规方法数据库访问对象代码具有更高的可重用性与可维护性。

常规方法与 ERS 方法性能比较结果如表 2 所示。

比较发现,ERS 方法在具有代码质量优势的同时,执行效率还要略快一些。其中原因是:执行查询任务时,超过 99%的时间耗费在数据库查询上,常规方法与 ERS 方法进行的数据库查询是完全相同的,因此总体耗时大致相同;但由于 ERS 方法拼接数据库查询语句的总次数较少,因此执行起来比常规方法就略快。

总之,试验显示:ERS 方法在明显提高可维护性与可重用性的同时,并没有降低系统的查询效率。

表 1 两种查询方法的代码矩阵比较

模块 代码	可维护 性指数	圈复 杂度	继承 深度	类耦合	代码 行数
数据访问对象 代码(常规方法)	73	16	1	10	27
资源库代码 (ERS 方法)	79	8	1	10	17

表 2 查询方法的性能比较

测试编号	测试方法	执行 10000 次耗时(ms)
第一次测试	常规方法	1820.41
	ERS 方法	1704.41
第二次测试	常规方法	1813.83
	ERS 方法	1696.28

7 结束语

文中提出扩展的资源库——规约方法(ERS 方法),并基于约束条件逻辑建立了规约继承体系。在通常情况下,ERS 方法比经优化的常规方法具有更高的可维护性、可重用性,运行效率也略高。

下一步研究工作的内容是把 ERS 方法与新型的数据访问方式——语言级集成查询(Language Integrated Query)相结合,进一步降低规约与具体数据库查询语句的耦合程度。

参考文献:

- [1] 林寒超,张南平. Hibernate 技术的研究[J]. 计算机技术与发展,2006,16(11):22-25.
- [2] Evans E, Fowler M. Specifications[C]//In: Proceedings of PLoP 97 Conference. USA: Washington University,1997.
- [3] Evans E. Domain-Driven Design: Tackling Complexity in the Heart of Software (Hardcover)[M]. USA: Addison - Wesley Professional,2003.
- [4] Fowler M, Rice D, Foemmel M, et al. Patterns of Enterprise Application Architecture[M]. USA: Addison - Wesley Professional,2003.
- [5] 唐金鹏,李玲琳. 持久层的设计与实现研究[J]. 企业技术开发,2008,4(4):12-14.

AppDB,然后把操作成功的信息用 AP 的公钥加密后返回给应用程序 AP,如果身份验证未通过则返回失败信息。

(10) 如果 AP 没有收到 Daemon 的回应消息,则表明通信失败,或者消息被拦截,则 AP 终止执行。如果 AP 收到 Daemon 的回应消息,则 AP 继续运行下去。

1.3 抗破解性能分析

破解方法有如下几种:

1) 截获 AP 和 Daemon 之间的通信消息,然后破解。

这个方法难度很大,这相当于破解公开密钥加密算法,此破解方法基本不可行。

2) 截获 Daemon 发送给 AP 的通信消息,把这个信息 OldMsg 保存起来,以后每次 AP 向 Daemon 发送请求都被破解进程 Cracker 截获并抛弃掉,然后把以前保留的消息 OldMsg 发送给 AP。

因为 AP 与 Daemon 的每次通信消息中都有一个流水号,每次通信的流水号都不相同,AP 收到消息后,首先把收到消息的流水号跟它发出请求消息中的流水号做比较,发现不同则终止该进程。此破解方法根本不可行。

3) 每次截获 AP 发送给 Daemon 的请求消息,并抛弃之。

AP 与 Daemon 之间是握手协议通信,对于 AP 的请求,Daemon 必须回应,若 Daemon 没有回应,则 AP 认定出现错误,并终止该进程。此破解方法也是根本不可行。

4) 备份这个系统的数据库文件 AppDB。

Daemon 进程每隔一个小时,从 AppDB 的 [main] 节中读取一个属性值 LastTime(时间戳),同时写进去一个新的时间戳 NowTime,如果 NowTime 和 LastTime 的时间差超过 MAX 小时,就表明有破解者把数据库文件替换了。这种防范策略有一个缺陷就是:用户的系统关机时间不能超过 MAX 小时,可以适当增大 MAX 的值来放宽限制,这使得破解者在 MAX 小时内的备份才有效。

5) 备份整个操作系统,从而达到恢复到原始状态的目的。

这种破解方法是可行的,但是,在操作系统被恢复后,还必须修改系统时间,否则 Daemon 进程肯定启动

不起来,正如第 4) 条的情况。

从以上分析可以得出如下结论:

(1) 缺点:这种防范策略可以被破解,尽管破解有一定的难度。

(2) 优点:该系统可以为多个软件提供软件版权的保护,而不必为每个软件做一个保护系统。

2 结束语

计算机网络的面世将人类带入了信息互联时代,也随之带来了计算机安全和网络安全问题,而最终的问题是软件的安全问题。因而软件保护技术一直是一个研究热点,它是以加密技术为基础的,而对于任何加密方法,必然有破解方法,不论采用何种保护方式,可执行程序都可以被黑客用 SoftICE 等工具调试跟踪,找到判断代码处,通过修改可执行文件,跳过此段代码,达到破解的目的,虽然采取一定的反跟踪、反调试技术,可以加大破解难度,但最终还是可以被破解。这使得对于软件版权的保护仅仅靠程序逻辑是不行的,必然要靠社会和法律来辅助保护。

参考文献:

- [1] 看雪. 加密与解密——软件保护技术及完全解决方案[M]. 北京:电子工业出版社,2001.
- [2] 周晓东,卢东明. 软件保护技术[M]. 北京:清华大学出版社,1994.
- [3] 郭勇,孔宝根. 软件保护及破解策略[J]. 航空维修与工程,2004(3):37-39.
- [4] 罗宏,蒋剑琴,曾庆凯. 用于软件保护的代码混淆技术[J]. 计算机工程,2006(11):177-179.
- [5] 祁明,容叶飞. 一种新型软件保护方法的设计与分析[J]. 微型电脑应用,2000(12):8-10.
- [6] 王珊,萨师煊. 数据库系统概论[M]. 北京:高等教育出版社,2006.
- [7] Peltier T R, Raton B. Information security policies and procedures: a practitioner's reference[M]. Florida: Auerbach Publications, 2004.
- [8] Garrett P, Lieman D. Public-key Cryptography[J]. American Mathematical Society, 2003(5):13-14.
- [9] 周玉洁,冯登国. 公开密钥密码算法及其快速实现[M]. 北京:国防工业出版社,2002.
- [10] Saloma A. 公钥密码学[M]. 丁存生,单炜娟译. 北京:国防工业出版社,1998.

(上接第 12 页)

- [6] 薛冰,营作良. 设计模式和数据持久层框架在 Web 系统中的应用[J]. 天津理工学院学报,2004,20(1):76-78.
- [7] 田志魏. 对象/关系映射持久化技术的研究及应用[J]. 微

计算机信息,2008,3(3):177-178.

- [8] 蒋科. 面向对象技术中 OR 映射框架的研究与应用[J]. 计算机技术与发展,2007,17(2):59-62.