一种软件设计模式的自动选择方法

钟金琴[1],辜丽川[2],孟 浩[2]
(1.安徽大学 电子与信息系,安徽 合肥 230011;
2.安徽农业大学 计算机学院,安徽 合肥 230036)

摘 要:设计模式是人们在实践过程中总结出来的成功设计范例,它的正确选择和使用是发挥模式作用的关键。而改变在设计模式应用过程中过分依靠人工的现状,有着重大的意义。重新应用需求和设计模式的形式化描述,在模式库中找出一组包含新应用领域的模式的可重用的设计以及重用模式的变换适配等方面,详细讨论了一种软件设计模式的自动选择方法。并结合实例给出了这种方法在网上证券交易原型系统中的实现。该方法为设计模式的重用提供了一个有效的途径,形式化描述是其基础,设计模式的组织和检索是其核心。利用此方法选择设计模式,可大大提高模式重用的准确性和软件开发的效率。

关键词:设计模式;BPSL;自动选择
中图分类号:TP311.5 **文献标识码**:A **文章编号**:1673-629X(2007)09-0021-04

A Method of Automatic Selection for Software Design Patterns

ZHONG Jin-qin[1],GU Li-chuan[2],MENG Hao[2]
(1.Department of Electronic and Information,Anhui University,Hefei 230011,China;
2.School of Computer,Anhui Agricultural University,Hefei 230036,China)

Abstract:Design patterns are successful design examples which people summarized in practice. How to correctly select and use these patterns are important to bring them into play. It's a great meaning to change the current situation that design pattern reuse depends on artificial operation excessively. This paper introduces a method of automatic selection for softeware design patterns from the respects, formally describing new application requirements and design patterns, finding out a group of reusable designs from pattern base in which new application fields are included, and reusing found design patterns by transformation. Then discusses the implementation of pattern selecting automaftically in the stock exchange system as an example. The base of method is formal descriptions, and the core is the organization and the selection of patterns' reusable design. The accuracy of design pattern reuse and the developing efficiency can be improved by using the method to select design pattern.

Key words:design pattern;BPSL;automatic selection

0 引 言

随着软件系统规模和复杂度的增加,每个软件开发都从头做起,会浪费很多的资源,因此软件重用逐渐成为软件界追求的目标。为了提高软件的重用效率、降低开发成本,软件专家引入了结构化、可重用的软件设计模式来捕捉并描述软件业多年来成熟的软件知识和经验,使人们在不同的应用领域中,可以开发出大粒度可重用的软件框架和组件,而不仅仅是简单地提供源代码。但设计模式的种类日益增多,相对于1995年Gang of Four (GoF)提出的23种通用的设计模式,设计模式的数量已经大大增加了。要从如此多的模式中选择适合自己系统的模式并非易事,选择正确、恰当的模式成为人们使用模式的瓶颈,尤其是对于模式不够熟悉的用户。因此,寻找一种简易有效的模式选择方法对于使用模式的用户来说非常重要。

收稿日期:2006-11-25
基金项目:安徽省高校青年教师科研资助计划项目(2006jql130);安徽省自然科学基金项目(2006KJ074B)
作者简介:钟金琴(1973-),女,安徽六安人,硕士,讲师,研究方向为软件工程。

1 设计模式的自动选择方法

传统的设计模式使用方法是使用者凭借对设计模式功能的了解和自身的设计经验,选择适当的设计模式进行软件开发。这种纯人工的方法已不能满足现实的需要。为了使用户在软件开发中能自动地选择正确的设计模式,近年来出现了一些重用部件的规格匹配方法。文中研究的这种设计模式的自动选择方法,也

是以此为基础的。模式引入软件开发首先需要建立一个模式库，系统设计员在模式库中选择可用的模式，以和需要解决的问题相匹配。通过对已有的模式选择方法加以总结、归纳、简化并把它们联系起来，发现在自动选择模式时基本可以遵循以下的步骤和原则：

(1)理解问题需求：明确需求的具体细节和所要达到的性能要求。按面向对象的分析方法给出每一功能模块的基本类、对象、方法及属性。并用形式化的语言对功能进行描述。

(2)匹配选取模式：根据模式的自动选择算法，从模式库中找出一组包含新应用领域的可重用的设计。这时可能会找出和问题相关的一个或多个设计模式。

(3)重用模式的变换适配：在此过程中，主要考虑设计模式在设计中所支持的可变化因素，即确定改变什么而不需重新设计，根据这一点可以找到所需的设计模式。此外考虑与其相关的设计模式。

1.1 设计模式的形式化描述

形式化描述方法是以严格的数学为基础，其优点是：表示严格，形式化，面向计算机。对设计模式的形式化描述是实现自动选择的前提。目前国内外已出现了一些对设计模式的形式化描述方法，如 Alencar 提出设计模式定义和应用的一种形式化方法、Eden 等提出一种叫做 LePUS 的形式化语言[1]、Tommi Mikkonen 给出了一种设计模式形式化规格说明语言 Disco 语言[2]。但这些方法在描述设计模式时，都存在着一个主要的问题，即缺乏完整性，一般只对设计模式的某些方面进行深入描述。在本方法中，选用了一个统一的完整的形式化描述语言 BPSL[3]。BPSL 语言弥补了以上语言的缺点，它把设计模式的结构和行为相结合，统一完整地实现了对设计模式的描述。

使用 BPSL 语言对设计模式进行描述，可分成四个部分：实体说明部分(即变量的声明及其类型说明)、持久关系描述部分、功能规范部分、行为语义部分。

(1)实体说明部分：对设计模式中涉及的类，以及类的属性、方法、对象等进行的描述。

(2)持久关系描述部分：这部分主要是对设计模式的结构部分进行描述。

(3)功能规范部分：对设计模式的功能本质进行抽象和形式化描述，其中不涉及设计模式的具体结构和行为，它只描述该设计模式能够为用户提供什么，而如何使用类的定义、类之间具有何种关系及使用哪些行为操作完成所需的功能不在描述范围内，其内容主要为传统设计模式描述中的 intent(意图)叙述部分，功能规范的表示使用前后置条件的方法。

(4)行为语义部分：对设计模式中存在的行为、操作的功能进行的形式化描述。

BPSL 中描述的类、属性、方法、对象等直接来源于面向对象的技术概念，因此容易被设计者所接受，使设计模式自动选择成为可能。

1.2 设计模式的自动选择算法

为了实现设计模式的自动选择，结合了形式化检索算法[4]和重用部件的规格匹配方法[5,6]，形成了适合于设计模式的自动选择方法——基于谓词匹配的设计模式选择算法。

定义 1：设计模式的形式化定义：设计模式(DP)的定义为 DP = < DPentity, DPprelation, DPfunction, DPaction >，其中 DPentity, DPprelation, DPfunction, DPaction 分别为 DP 的实体说明部分、持久关系部分、功能规范部分和行为语义部分。

定义 2：设计模式间的匹配关系：match(DP, DP') = matchentity ∧ DPentity, DP' entity) ∧ matchprelation (DPprelation, DP' prelation) ∧ matchfunction (DPfunction, DP' function) ∧ matchaction(DPaction, DP' action)

由于设计模式的实体关系及持久关系部分比较复杂，行为语义部分变化多端，所以不容易进行匹配，在这里主要实现功能规范匹配。定义如下：

matchfunction(DPfunction, DP' function) = ((DPpre(DPfunction)↔DP' pre(DP' function)) ∧ (DPpost(DPfunction)↔DP' post (DP' function))) ∨ ((DPpre(DPfunction)→DP' pre(DP' function)) ∧ (DPpost(DPfunction)←DP' post(DP' function)))

即表明如果两个设计模式的功能规范部分的前后置条件都等价，就认为这两个功能规范部分等价；如果 DP' function 中的前置条件比 DPfunction 中的前置条件强，而 DP' function 中后置条件比 DPfunction 中的后置条件弱，则认为这两个功能规范相似。文中因功能规范的前后置条件是通过谓词的连接来描述的，在匹配时要求对谓词名称和谓词变量的类型(C, A, M, O, V)进行匹配。对某一谓词来说，若谓词名称相同，且谓词变量的个数及类型相同，则认为匹配成功。若前后置条件中所有的谓词匹配成功，则认为此问题可用模式库中的该模式来解决。

定义 3：设计模式的选择定义：retrieve(Q, matchfunction, L) = {DP∈ L: matchfunction(DP, Q)}，其中 Q 为需求规范，L 为设计模式集合，DP 为设计模式。

本算法是以形式化描述为前提，消除了语义歧义性。并提出一种包涵匹配思想，在对查询的模式不完全描述的情况下，查询匹配具备很强的松弛能力，可以在保证较高查准率的前提下兼具较高的查全率。另外，通过利用动态规划和合理设定约束条件，不仅能给用户返回匹配结果，还能反映出匹配的程度，从而保证

模式查询具有足够的查询效率,为用户复用模式提供了有意义的参考信息。

2 设计模式自动选择方法的应用

在网上证券交易原型系统的开发设计过程中,使用了这种设计模式的自动选择方法,使开发效率有了大幅度的提高。下面选择证券交易原型系统行情管理子系统的设计与实现来说明设计模式重用方法。

2.1 理解问题需求

行情管理模块是证券交易原型系统中一个重要的功能模块。其主要功能是接收客户查询行情的请求,根据请求的不同分别返回:实时行情、个股当日分钟行情、个股当日分价行情、个股历史数据和个股当日成交明细数据。

就行情数据而言,整个系统是一种"瘦客户,胖服务器"的构架,即客户端不存放任何数据,而是根据需要向服务器端请求。若服务器端数据发生改变,客户端观察到的数据也随之改变。因此根据对需求的分析,得到两个类:一个是分派请求类 hqrequest,另一个为获取行情类 hqresponse。其中 hqresponse 具有的方法为:与分派请求建立通信 Attach(),通报分派请求类数据的改变 Notify(),与具体的分派请求撤销通信 distach()。hqrequest 具有的方法为:接收获取行情类的通报后相应地更新数据 update()。用 BPSL 语言形式化描述如下:

实体描述:

hqresponse,hqrequest∈C;

hqresponse－data,hqrequest－data∈A;

attach,Notify,distach,update∈M;

concrete－hqresponse,concrete－hqrequest∈O;

m,n∈V;

功能规范描述:

前置条件:

defined－in(m,hqresponse) (defined－in(n,hqrequest);

后置条件:

equal(hqresonse－data,hqrequest－data);

其次在获取行情数据时,行情数据又分为五种:明细、实时、分钟、历史和分价行情。而这五种行情之间存在着一定的相似关系,在获取这五种行情的时候,都会用到一些相同的操作,比如:解析数据,即从一个数据包的字节中提取出有用的行情信息。因此在设计时可以为它们定义一个公共的父类行情信息类 hqdata,但由于不同行情信息的结构不同,使得各自解析数据的方法也必定不同。为此在父类中定义一个抽象方法 parsedata(),而延迟到子类中实现。

实体描述:

Hqdata,mshqdata,sshqdata,fzhqdata,lshqdata,fjhqdata∈C;

parsedata∈M;

concrete－hqdata,concrete－mshqdata,concrete－sshqdata,concrete－fzhqdata,concrete－lshqdata,conrete－fjhqdata∈O;

功能规范描述:

前置条件:

defined－in(parsedata,hqdata);

后置条件:

Inheritance(mshqdata,hqdata) ∧ Inheritance(sshqdata,hqdata) ∧ Inheritance(fzhqdata,hqdata) ∧ Inheritance(lshqdata,hqdata) ∧ Inheritance(fjhqdata,hqdata)

另外在获取行情时,需要从数据模块中得到模拟的行情数据。两者之间采用 UDP 包进行通信。获取行情模块收到 UDP 报文后,需要将之规范成为一个行情数据格式的结构。这种规范算法可能根据 UDP 报文文体定义的改变而改变。因此在设计时可以将五种类型行情数据的格式化算法进行封装,作为一个独立的类,对外只表现为一个输入口:UDP 报文和一个输出口:行情数据结构。这样维护一个独立类的对象,用这个对象支处理收到的 UDP 报文,当 UDP 报文结构发生变化而需要修改算法时,只需要用另外的一个具体算法对象替换原有的就可以了,而无须改变其他的类或对象。

实体描述:

Gethqdata,formalgorithm∈C;

dataformat∈M;

Concrete－hqdata,concrete－formalgorithm∈O;

功能规范描述:

前置条件:

defined－in(1,gethqdata) ∧ defined－in(5,formalgorithm);

后置条件:

Reference－to－many(gethqdata,formalgorithm)。

2.2 匹配选取模式

根据以上分析得到功能规范的形式化描述,采用基于谓词匹配的设计模式的自动选择算法,从模式库中检索出适合要求的设计模式。文中使用 Delphi7.0 及 ModelMaker6.2 来实现模式的匹配检索,数据库是以 SQL Server2000 为后台的。图 1～3 为行情管理模块的设计模式自动选择过程。

2.3 重用模式的变换适配

在实际设计中,由于模式间存在一定的联系,某些模式非常相似,这就使得在模式匹配过程中,可能会出现几个模式都能解决某一问题,这时就需要对检索出来的模式再裁剪,目前还需要人为处理这一问题。另外,对已确定的某一模式能解决某一问题的情况下,也

经常会发现要对这一模式进行修改变换后才能适用特定的系统。ModelMaker 是一种智能模式工具，它可以实现重用模式的修改变换，以适用特定的系统。如图 3 所示。

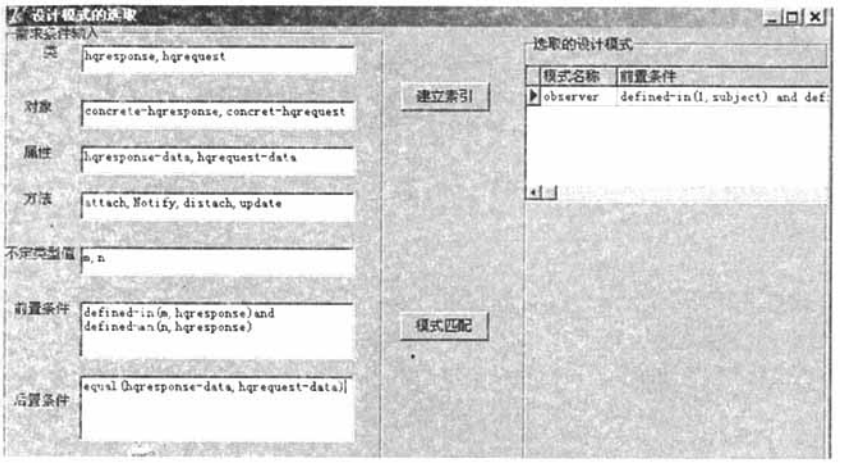

图 1 设计模式的选取界面

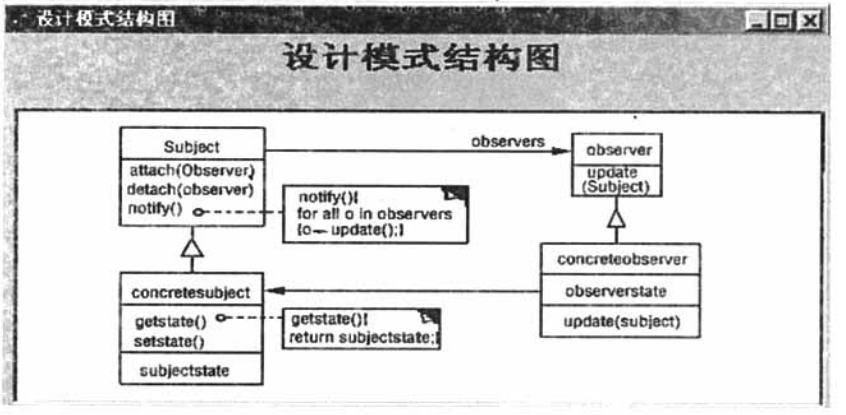

图 2 提供模式信息界面

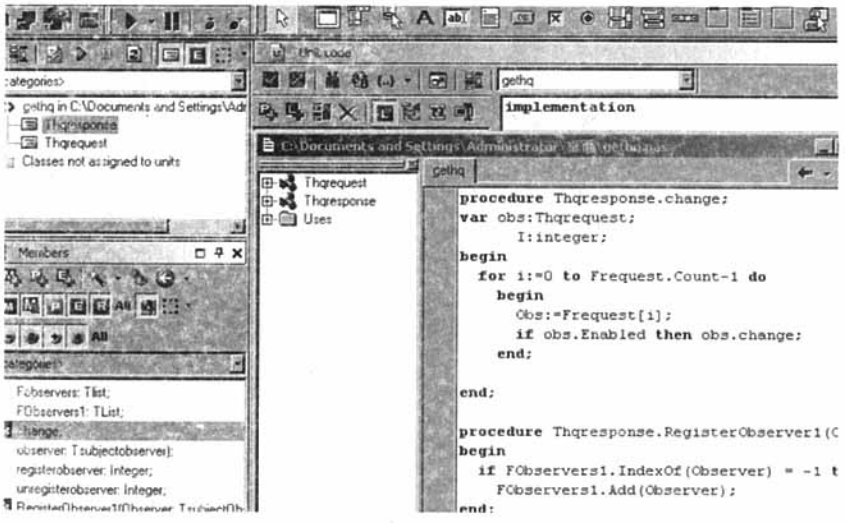

图 3 模式的应用

对其他的功能模块采用同样的方法，就能较好地实现模式的重用。

3 小 结

本方法提出的根本目的就是要把设计模式施用到软件设计过程，有效地指导软件开发。有工具的支持能避免失误，使描述更为精确，能快速完成编码，并在不同的实现中保持一致性。更重要的是，它支持程序的深化（如与已有结构的集成、修改和扩充已有的结构、类和程序等）且表达能力更强。但方法运用的工具还未能完全实现，还需要进行研究。

参考文献：

[1] Eden A H, Gil J, Hirshfeld Y, et al. Towards a Mathematical Foundation for Design Patterns [EB/OL]. 2000 - 08. http://www.math.tau.ac.il/eden/bibliogrphy.html.

[2] Katara M. Aspects of Continuous Behaviour - Design of Real - time Reactive Systems in DisCo[D]. Tampere: Tampere University of Technology, 2001.

[3] Taibi T, Check D, Ngo L. Formal Specification of Design Patterns - A Balanced Approach[J]. in Journal of Object Technology, 2003, 2(4): 127 - 140.

[4] Zaremski A M, Wing J M. Signature Matching: a Key to Reuse[J]. ACM Sigsoft Software Engineering Notes, 1993, 18(5): 182 - 190.

[5] Chang Chao - Tsun, Chu W

（下转第 27 页）

5　算法仿真

根据上面提出的匹配算法,对其进行了计算机仿真实验。选择了适于匹配的地形图如图 1 和图 2 所示,设定仿真试验的条件,INS 的初始误差为 0.02,测深测潜仪的量测误差为标准量测误差,电子海图的制图误差为 5 米,仿真步长 14 秒,陀螺仪参数为 0.003 度,随机游走为 0.01 度。

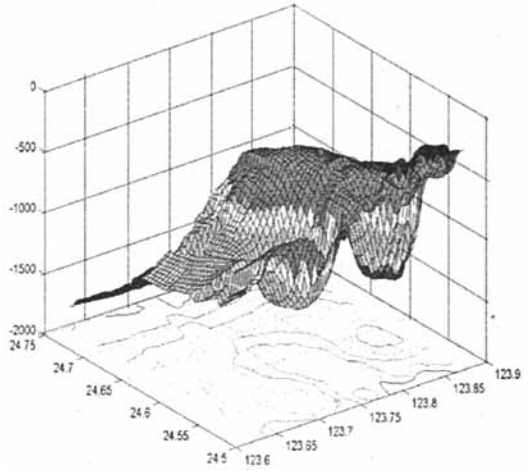

图 1　地形 5 适于匹配高程三维图

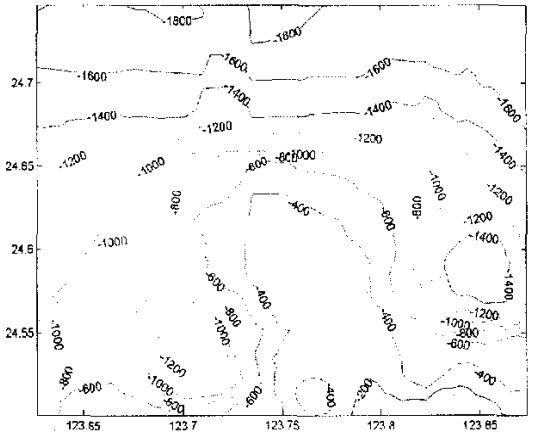

图 2　适于匹配高程等高线图

从试验结果图 3～4 来看,地形熵和地形差异熵综合匹配算法能够将匹配平均经纬度误差控制在 200 米以内,圆概率误差值能控制在 250 米以内,因此,该算法具有较好的收敛匹配误差的作用。

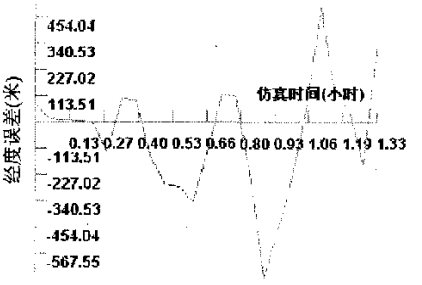

图 3　算法匹配后的经度误差图

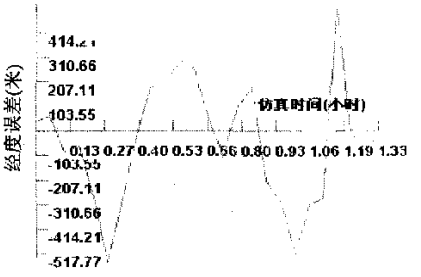

图 4　算法匹配后的纬度误差图

6　结　论

文中探讨并设计了基于地形熵和地形差异熵的综合地形匹配算法,并利用 VC++语言实现了该算法,通过大量实验研究表明,该算法符合匹配精度和速度的要求,基本能够完成水下运载体辅助导航的任务。

参考文献:

[1]　马洪波,刘建辉,杨　健.基于地形熵差和高程绝对差度量的地形匹配算法[J]. 指挥技术学院学报,2000(5):60－63.

[2]　刘光军,陈　晶.海底地形辅助导航系统仿真技术研究[J].计算机仿真,2000(2):21－23.

[3]　张飞舟.水下无源导航系统仿真匹配算法研究[J].武汉大学学报,2003(2):33－35.

[4]　李爱军,白　焱.高程熵综合地形匹配算法的仿真研究[J].计算机仿真,2004(8):135－137.

[5]　刘光军,袁书民.海底地形匹配技术研究[J].中国惯性技术学报,1999(1):20－22.

[6]　Bar－Gill A. Improvement of Terrain－Aided Navigation Via Trajectory Optimization[J]. IEEE Trans on Control Systems Technology,1994(4):336－342.

(上接第 24 页)

C,Liu Chung－Shyan,et al. A Formal Approach to Software Components Classification and Retrieval[C]//Proceedings of the COMPSAC'97－21st International Computer Software and Applications Conference. Washington D C:[s.n.],1997.

[6]　万剑怡,薛锦云.使用规范匹配实现设计模式的自动获取[J].小型微型计算机系统,2003,24(3):326－329.