

可拓策略生成系统的构件化设计与实现

叶广仔¹, 李卫华², 李淑飞¹

(1. 东莞职业技术学院 计算机工程系, 广东 东莞, 523808; 2. 广东工业大学 计算机学院, 广东 广州, 510006)

摘要:将可拓学的基本理论和可拓工程的方法运用到防治空气污染的策略生成当中,并通过计算机实现了关于地区性二氧化硫浓度矛盾问题的可拓策略生成系统.系统利用面向构件的开发技术,通过Java EE 5架构,把JSP、Servlet、EJB等技术相结合,大大增加了系统的可重用性和可维护性.

关键词:可拓学;策略生成;构件化

中图分类号: TP30 **文献标识码:** A **文章编号:** 1673-4785(2010)04-0366-06

The design and implementation of extension strategy Generation system based on components

YE Guang-zai¹, LI Wei-hua², LI Shu-fei¹

(1. Department of Computer Engineering, Dongguan Polytechnic, Dongguan 523808, China; 2. Guangdong University of Technology, Guangzhou 510006, China)

Abstract: Applying the basic theory of the Extenics and the method of extension project to the strategy generating system for air pollution prevention. It presents the process of implement the extension strategy generating system on computer to solve the regional SO₂ concentration contradiction problem and shows the running results. The system applies the component-oriented development of technology. Through the platform of Java EE 5, it combines the JSP, Servlet, EJB, and other technologies, and greatly increases its reusability and maintainability.

Keywords: extenics; strategy generating; componentization

可拓策略生成系统^[1] (ESGS)把可拓技术和现有的人工智能技术、数据库技术、可视化技术等相结合,为决策者提供合适的方案,是一种解决不相容问题^[2]的工具.但可拓策略生成系统的实现还处于初期阶段,要实现一个比较完善的,可重用性和可维护性高的可拓策略生成系统是当前研究的重点.因此,引入以构件化的开发方式来实现可拓策略生成系统的各业务流程组件,在开发过程中,使用Java EE 5构件,以面向构件的软件开发方法,来实现可拓策略生成系统核问题模型的建立模块、策略生成模块、以及策略评价模块,使系统具有更大的灵活性和复用性,使其能通过构件的替换来满足不同用户的需求和业务功能的变化需求.

1 可拓策略生成方法

可拓策略生成方法^[3]是以可拓学^[4]为理论基础,

模拟人类的思维模式,用形式化、定量化方式生成解决不相容问题的策略的一种方法.它基本思路是:

1) 建立核问题模型

通过人-机对话界定不相容问题的目标、条件以及一些衡量参数:如最优值、量变区域和质变区域等指标,然后用基元^[5]表示体系建立核问题的可拓模型.

2) 核问题相容度计算

建立关联函数,通过计算,判定核问题不相容的程度.相容度如果为非负数则问题相容;如果为负数则问题不相容.

3) 不相容问题的拓展分析

对不相容核问题模型的条件基元进行相关分析,建立问题的相关树^[1];然后对相关树的树叶(基元)进行发散分析.

4) 问题的可拓变换

对发散分析后的叶子节点进行置换变换,再根据传导变换,形成传导变换的蕴含树.由可拓变换和传导变换形成的树,通常称为可拓策略生成树.

5)策略的优度计算

对变换后的问题再次进行相容度计算,若其相容度由负数变成非负数,则此可拓变换或者变换的运算为解决不相容问题的可拓策略.最后根据优度的大小,正数值越大策略的优度越高,从而取得较优的策略.

2 可拓策略生成系统功能模块的构件化设计

本文所实现的可拓策略生成系统有以下特点:

- 1) 系统使用的设计模式为 MVC 模式:以 EJB 组件实现应用的业务逻辑,由 JSP 页面产生应用的表示面,由 Servlet 提供应用的处理过程控制.
- 2) 本系统基于构件来实现,根据系统的业务逻辑划分系统的功能模块,主要使用 EJB3.0^[6] 中的 Entity Bean 来实现数据库中表的建立及维护;并通过 EJB3.0 中 Stateless Session Bean 及 Stateful Session Bean 来实现系统的业务功能模块.

2.1 系统的四层架构设计

本文通过 Java EE 5 架构^[7],实现一个基于构件的可拓策略生成系统.这一系统使用 MVC 的开发模式,把 JSP、Servlet、EJB 等技术结合起来.通过数据库中的表与 Entity Bean 相互映射,Entity Bean 的操作实现对数据库中表的操作.如图 1 所示,描述了本系统 4 层结构 (Application Client Container、Web Container、EJB Container、Database) 的 B/S 架构下通过浏览器访问数据库的流程.

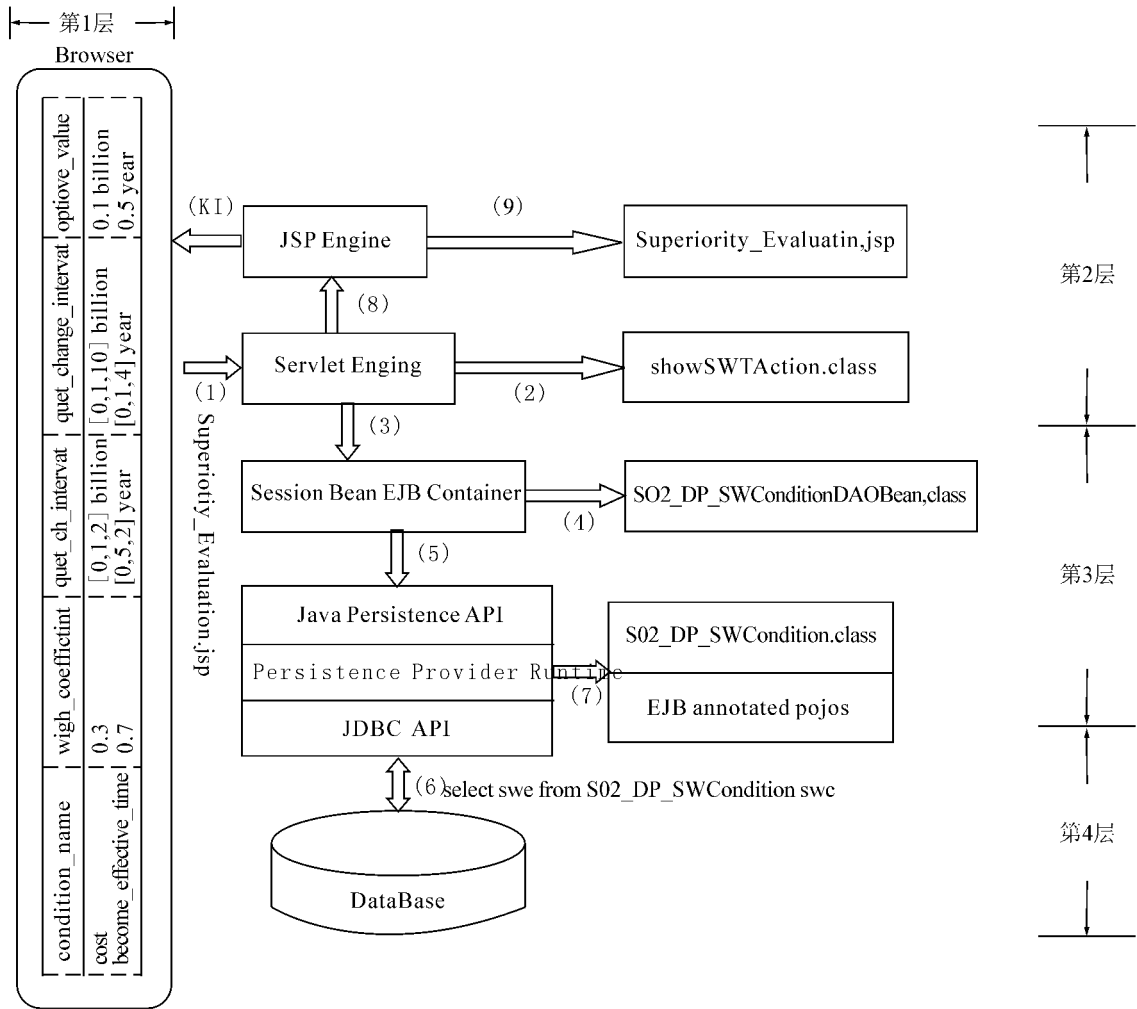


图 1 浏览器访问数据库中表的过程图
Fig. 1 The process of Browser accessing database table

2.2 使用 Entity Bean 实现类基元

下面以衡量条件类基元为例,介绍通过 Entity Bean 实现类基元的方法:
衡量条件类基元是衡量策略优度的计算中用到

的一个重要概念.根据对衡量条件的计算可以实现策略的优度评价,从而得出策略中优度比较高的策略.衡量条件类基元如图 2 所示:

$$\{B\} = (\{O\}, C, V) = \begin{bmatrix} \{O\} & c_1 & v_1 \\ & c_2 & v_2 \\ & \vdots & \vdots \\ & c_n & v_n \end{bmatrix} = \begin{bmatrix} \text{condition_name} & \text{weigh_coefficient} & v_1 \\ & \text{quan_ch_interval} & v_2 \\ & \text{qual_chang_interval} & v_3 \\ & \text{optimum_value} & v_4 \end{bmatrix}.$$

图2 衡量条件类基元

Fig. 2 The basic-element of weigh condition class

其中, $\{O\}$ 表示一类对象(物, 动作或关系); c_1, c_2, \dots, c_n 表示对象 O 的 n 个特征, v_1, v_2, \dots, v_n 表示

对象 O 关于上述特征的相应量值, 它对应数据库中的表与 EJB3.0 构件中的实体 Bean 如图3所示:

mysql>describe s02_DP_SWCondition:						12/**	
						13*@author root S02浓度问题衡量条件表	
						14*/	
						15@Entity	
						16@Table(name="S02_DP_SWCondition")	
						17public class S02_DP_SWCondition implements Serializable{	
						18 /**	
						19 *	
						20 */	
						21 private static final long serialVersionUID=1L; //衡量条件名称	
						22 private String condition_name;	
						23 private String weigh_coefficient;	
						24 private String quan_ch_interval;	
						25 private String qual_change_interval;	
						26 private String optimum_value;	
						??	
						Entity Bean	

图3 S02_DP_SWCondition 表与 Entity Bean 间的映射图

Fig. 3 The Mapping of the S02_DP_SWCondition table and the Entity Bean

2.3 通过 Session Bean 实现业务功能

可拓策略生成系统中,经常要计算问题相容度,特别是计算核问题的相容度以及策略的优劣.因此,本系统实现了一个基于初等关联函数的相关度计算构件.下面以初等关联函数的计算构件为例,讲述了通过无状态会话 Bean 来实现初等关联函数的计算过程.

1. 初等关联函数,主要公式如下:

1) 文献[3]中提到,根据量值的最优值 x_0 在 X_0 中的位置,应选择对应的测距计算公式,譬如,当 $X_0 = \langle a, b \rangle, x_0 = a$,则 x 与区间 X_0 的测距公式有:

$$\rho_l(x, x_0, X_0) = \begin{cases} a - x, & x < a, \\ a_x, & x = a, \\ x - b, & x > a. \end{cases} \quad (1)$$

式中:

$$a_x = \rho_l(a, a, X_0) = \begin{cases} 0, & a \notin X_0 \\ a - b, & a \in X_0, \\ 0 \otimes (a - b), & a \notin X_0 \text{ 且 } a \in X_0. \end{cases}$$

关于最优值 $x_0 \in (a, \frac{a+b}{2}), x_0 \in (\frac{a+b}{2}, b)$ 或者 $x_0 = b$ 等情况,详细请看文献[3].

2) 一个点与2个区间的位置关系.一般地,设 $X_0 = \langle a, b \rangle, X = \langle c, d \rangle$, 且 $X_0 \subset X$, 则点 x 关于

区间 X_0 和 X 组成的区间套的位值^[3] 规定为

$$D(x, X_0, X) =$$

$$\begin{cases} \rho(x, X) - \rho(x, X_0), & \rho(x, X) \neq \rho(x, X_0) \text{ 且 } x \notin X_0, \\ \rho(x, X) - \rho(x, X_0) + a - b, & \rho(x, X) \neq \rho(x, X_0) \text{ 且 } x \in X_0, \\ a - b, & \rho(x, X) = \rho(x, X_0). \end{cases} \quad (2)$$

3) 设 x 为实轴上的任一点, $X_0 = \langle a, b \rangle$ 为实域上的任一区间,称

$$\rho(x, X_0) = \left| x - \frac{a+b}{2} \right| - \frac{b-a}{2}. \quad (3)$$

为点 x 与区间 X_0 之距^[1]. 其中 $\langle a, b \rangle$ 既可为开区间,也可为闭区间,也可为半开半闭区间.

4) 设 $X_0 = \langle a, b \rangle, X = \langle c, d \rangle, X_0 \subset X$, 且无公共端点,建立初等关联函数^[3]:

$$k(x) = \frac{\rho(x, x_0, X_0)}{D(x, X_0, X)}. \quad (4)$$

与以上的4个公式对应的 Session Bean 实现如图4和图5所示,它实现了初等关联函数的主要计算.即“求解测距”函数实现公式1)的计算,“计算位值”函数实现式2)的计算,“计算点与区间的距”实现式(3)的计算,“计算关联度”实现式(4)的计算.

其中 DepFunDAO 为 DepFunDAOBean 的接口,在 EJB 3.0 应用中,藕合松散的服务构件的典型应用是 Session Bean. 一个 Session Bean 至少有一个接口,其

它应用构件通过它获得服务. 其中注释“@ Remote”表示它是一个远程接口, 可以实现远程访问. “@ Stateless”表示此 Session Bean 为无状态会话 Bean, 对于无状态(Stateless)的 Session Bean, 每次的请求都是被随机挑选的 Session Bean 实例来处理的.

```

1 package slsb;
2
3 import javax.ejb.Remote;
4
5 @Remote
6 public interface DepFunDAO {
7     // 计算函数
8     * s1为计算函数, s2为计算函数, x为计算函数, x0为计算函数
9     * flag1为true表示区间左端点为0, flag2为true表示区间右端点为1
10    * a, b分别为计算函数区间的左端点, c, d分别为计算函数区间的右端点
11    * l为计算函数区间的左端点, r为计算函数区间的右端点
12
13    // 计算函数
14    public float p(float x, float a, float b, float x0, boolean flag1, boolean flag2);
15    // 计算函数
16    public float d(float x, float a, float b, float c, float d);
17    // 计算函数
18    public float p(float x, float l, float r);
19    // 计算函数
20    public float PriDepFunction(float x, String s1, String s2, float x0);
21
22 }

```

图4 DepFunDAO 接口的实现

Fig. 4 The realization of DepFunDAO interface

```

1 package slsb;
2
3 import javax.ejb.Stateless;
4 import slsb.DepFunDAO;
5
6 @Stateless
7 public class DepFunDAOBean implements DepFunDAO {
8     // 计算函数
9     public float p(float x, float a, float b, float x0, boolean flag1, boolean flag2) {
10         float p = 0;
11         if (x0 == a) {
12             if (x < a) {
13                 p = a - x;
14             } else if (x > a) {
15                 p = x - a;
16             } else {
17                 if (flag1) {
18                     p = a - b;
19                 } else {
20                     p = 0;
21                 }
22             }
23         }
24     }
25 }

```

图5 DepFunDAOBean 的实现

Fig. 5 The realization of DepFunDAOBean

3 构件化可拓策略生成系统的实现

本系统开发的总体任务是通过使用 Java EE 5 框架, 利用面向构件的开发方式实现一个可拓策略生成的系统. 下面介绍系统开发的环境、系统业务逻辑功能及演示效果图:

3.1 系统开发环境

本系统的开发环境为

操作系统: Linux 2.6.21-1.3194.fc7 (i386);

开发平台: eclipse3.2^{[8][9]};

JDK 版本: Java SE 6;

数据库服务器: Mysql5.0;

应用服务器 Jboss4.2^[10].

3.2 系统功能

本系统的主要业务功能划分如下

- 1) 用户登陆及验证;
- 2) 核问题模型的建立;
- 3) 通过拓展分析与可拓变换生成策略;
- 4) 对策略进行优度评价.

3.3 演示效果图

某地区 A 的现今大气 SO₂ 浓度是 0.07 mg/L, 当地政府部门要求把地区的 SO₂ 浓度控制在 [0.02, 0.06] mg/L 这个范围之内, 显然这是一个主观愿望与客观条件间的主客观矛盾问题, 下面以降低地区 SO₂ 浓度的策略生成过程为例, 介绍本系统策略生成的过程.

1) 登录系统后, 进入到问题模型界面, 如图 6 所示. 用户可以通过下拉列表选择要解决的表层问题. 这里以污染问题为例, 当用户选择“pollution problem”提交后, 如图 7 所示, 用户就可以看见关于所选表层问题的核问题选择下拉列表. 以 SO₂ 空气污染问题为例, 当用户选择“SO₂_pollution_problem”提交后, 如图 8 所示, 界面中会显示关于所选核问题的一些默认参数: 其中, 如果核问题为: $P_0 = g_0 * I_0 = (Z_0, c_{0s}, X_0) * (Z_0, c_{0t}, c_{0e}(Z_0))$, 则 Np_Zo 表示 Z_0 , Np_Cos 表示 c_{0s} , Np_cot 表示 c_{0e} , Current_condition 表示 $c_{0e}(Z_0)$, Quan_ch_interval 表示 X_0 ;

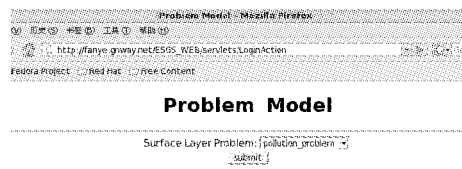


图6 表层问题选择界面

Fig. 6 Surface layer problem select interface

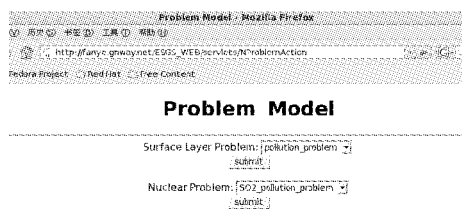


图7 核问题选择界面

Fig. 7 Kernel problem select interface

另外, 因为本系统在计算问题相容度时使用初等关联函数, 即公式为 $k(x) = \rho(x, x_0, X_0) / D(x, X_0, X)$, 其中:

$x = \text{Current_condition}(\text{Unit})$, $x_0 = \text{Optimum_value}(\text{Unit})$,

$X_0 = \text{Quan_ch_interval}(\text{Unit})$, $X = \text{Qual_ch_interval}(\text{Unit})$.

2) 当设置核问题的参数(如图 8 所示)并提交

后,用户就可以得到核问题的可拓模型(如图9所示),及其相容度.如图7所示,其中,核问题的可拓模型是:

$p_{00} = g_{00} * l_{00} = (\text{area_A}, \text{SO2_density_target}, [0.02, 0.06] \text{ mg/L}) * (\text{area_A}, \text{SO2_density_condition}, 0.07 \text{ mg/L})$, 其相容度为 $K(p_{00}) = -0.125\ 000\ 03 < 0$. 由于核问题的相容度小于0,所以用户所设置的核问题为不相容问题,也就是矛盾问题.为了得到解决问题的策略,用户可以从拓展分析列表选取一种对核问题的拓展分析方法,本文以相关分析为例,选取 Correlation_Analysis. 提交后,就可以进入核问题的拓展分析与可拓变换页面(图10).

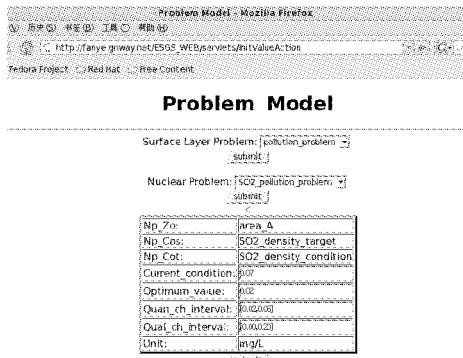


图8 核问题默认参数显示界面

Fig. 8 Kernel problem default parameters interface

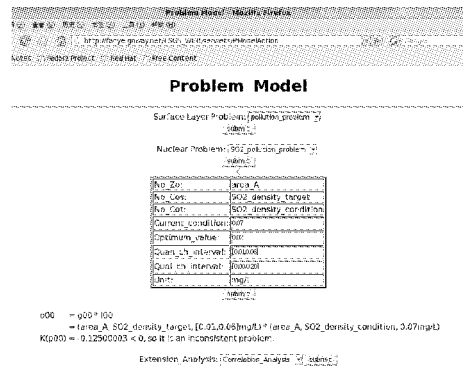


图9 核问题的可拓模型显示界面

Fig. 9 Kernel problem on the extension model display interface

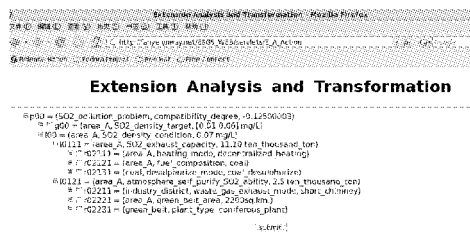


图10 相关分析界面

Fig. 10 Correlation analysis interface

3) 进入核问题相关分析界面(图10)后,用户可以看到针对核问题的初始状态相关树,其中各个节点都是一个基元.用户可以在初始状态相关树中选取叶子节点作发散分析,如图11所示,在发散分析树中,各个节点都有相应的下拉列表,它们分别对初始状态相关树中选取叶子节点进行发散分析的结果.用户可以在各个下拉列表中选择用于置换初始态的物元,根据可拓变换(Extension transformation)类型进行可拓变换,这里以置换变换(Replacement transformation)为例,提交后,用户可以得到针对核问题可拓变换后的状态树及其相容度,如图12所示.如果可拓变换后核问题的相容度大于0则此可拓变换能使核问题从不相容问题转换成相容问题,即此可拓变换为解决不相容核问题的策略.然后按优选度分析(Superiority evaluation)按钮进入优选度评价页面(如图13)所示.

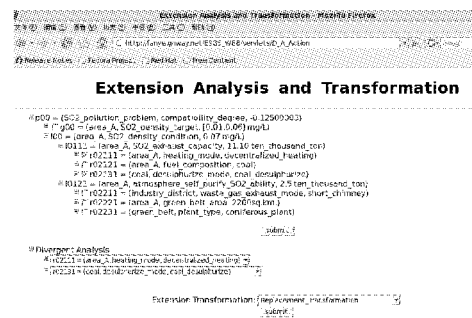


图11 置换变换界面

Fig. 11 Replacement transform interface

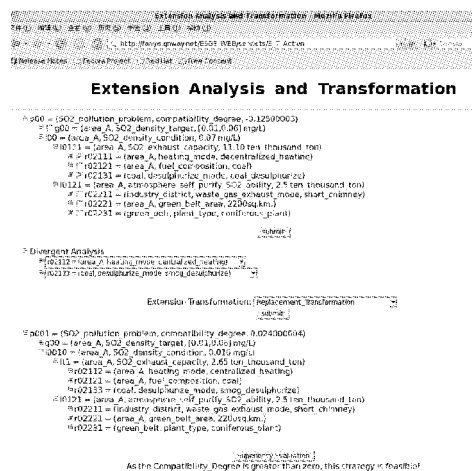


图12 策略生成界面

Fig. 12 Strategies generated interface

4) 进入优选度评价界面(图13),用户可以看到一个关于策略优选度评价的衡量条件表.用户可以根据实际情况设定策略的衡量条件参数值.在计算策略的优选度时,式(2)中的 x_0 等于 optimum_value 列中的

值, X_0 等于 quan_ch_interval 列中的值, X 等于 qual_ch_interval 列中的值. 式(4)中的 k 等于 weigh_coeficient 列中的值. 提交设置好的参数后, 用户就可以得到策略的优度值, 如图 14 所示. 当优度值大于 0 时, 策略可取, 值越大, 策略越优; 当优度值小于等于 0 时, 策略不可取, 值越小, 策略越差.

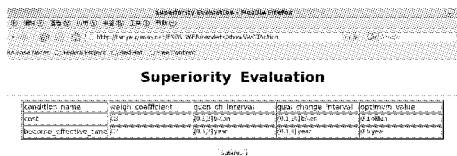


图 13 策略衡量条件参数设置界面

Fig. 13 Set of parameters to strategy weigh condition interface

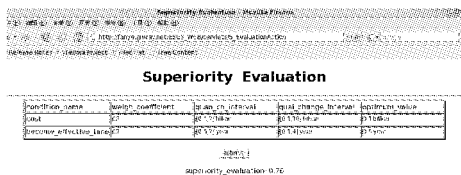


图 14 策略的优度评价界面

Fig. 14 Strategy superiority evaluation interface

4 结束语

正如文献[1]指出:“现在的人工智能的知识表现方法无法描述处理矛盾问题的过程, 然而, 可拓学与人工智能相结合, 并用计算机处理矛盾问题, 这对提高人类智能和机器智能都有重大的意义。”可拓策略生成系统的特点是根据可拓工程方法生成可以解决矛盾问题的策略, 本系统把可拓技术与人工智能技术、数据库技术、面向构件技术等多种技术结合在一起进行策略生成, 能够得到传统软件决策系统无法生成的可拓策略, 提供给人们解决空气污染问题时参考. 目前该系统是 B/S 架构, 基于 4 层结构 (Browser、Web Container、EJB Container、Database) 的可拓策略生成系统 (ESGS), 系统构件的可重用性和可维护性较好. 同时系统使用了 Java EE 5 架构, 因此也适合于中小系统的开发, 通过 EJB3.0, 可以容易地开发出所需构件, 有利于以后进一步完善可拓策略生成系统及其开发.

参考文献:

- [1] 李立希, 杨春燕, 李铎汶. 可拓策略生成系统[M]. 北京: 科学出版社, 2006: 2-146.
- [2] 蔡文. 可拓集合和不相容问题[J]. 科学探索学报, 1983, 3(1): 83-97.
- [3] 杨春燕, 蔡文. 可拓工程[M]. 北京: 科学出版社, 2007: P201-211.
- [4] 蔡文, 杨春燕, 何斌. 可拓逻辑初步[M]. 北京: 科学出版社, 2003: 133-138.
- [5] 蔡文, 杨春燕, 陈文伟, 等. 可拓集与可拓数据挖掘[M]. 北京: 科学出版社, 2008: 22-28.
- [6] 黎活明. EJB3.0 实例教程[M]. 北京: 北京传智播客教育科技有限公司, 2007: 87-204.
- [7] 李绪成, 腾英岩, 闫海珍. Java EE 5 实用教程-基于 WebLogic 和 Eclipse[M]. 北京: 电子工业出版社, 2007: 2-7.
- [8] 张大治, 应群. 精通 Eclipse[M]. 北京: 清华大学出版社, 2006: 271-412.
- [9] Berthold Daum. 李化, 李政仪, 译. Eclipse 3 高级编程[M]. 北京: 清华大学出版社, 2006: 115-171.
- [10] 张洪斌. JBoss 平台上的 Java EE 程序开发指南[M]. 北京: 清华大学出版社, 2007: 137-264.

作者简介:

叶广仔, 男, 1981 年生, 助教, 硕士研究生, 研究方向: 智能软件

李卫华, 女, 1957 年生, 教授, 博士, 研究方向: 面向 Agent 计算、网络信息系统、智能软件等

李淑飞, 女, 1974 年生, 讲师, 硕士, 研究方向: 计算机技术