

# 基于框架理论的防误系统可移植性提高方法

王友钊, 黄冬

(浙江大学数字技术及仪器研究所, 杭州 310027)

**摘 要:** 在线式微机防误系统存在可移植性不高的缺陷, 导致其在国内电力系统推广缓慢。为此, 提出一种基于框架理论的可移植性提高方法, 采用框架表示法描述微机防误系统的设备与逻辑参数, 构建一个易修改、移植的框架网络数据结构模型, 并对其进行优化与改进。通过框架理论与面向对象语言 Visual C++ 的融合, 实现基于该结构的微机防误系统软件。实验结果表明, 该方法使搜索时间缩短为 3.9 s, 搜索准确率提高到 99.5%, 能提升知识在计算机中存储、检索、使用和修改的效率, 节省微机防误系统移植的时间。

**关键词:** 框架理论; 框架网络; 微机防误系统; 逻辑判断; 网络维度; 可移植性

## Portability Improvement Method of Anti-misoperation System Based on Frame Theory

WANG You-zhao, HUANG Dong

(Institute of Advanced Digital Technology and Instrumentation, Zhejiang University, Hangzhou 310027, China)

**【Abstract】** The importability of online microcomputer anti-disoperation system makes it difficult to propagate in domestic electric power system. In view of problem that microcomputer anti-misoperation system portability is not high, this paper proposes a portability improvement method of microcomputer anti-misoperation system based on the frame theory. It describes the parameters and information of devices and logic used in the system, builds and optimizes a revisable and portable model of the frame network data structure. Through the combination of frame theory and object-oriented language Visual C++, it builds the software based on this structure. The outcome of tests shows that this method can shorten the search time to 3.9 s, and can improve search accuracy to 99.5%. It increases the efficiency of the storage, retrieval, application and modification of information in the computer, and saves the time needed for grafting the system.

**【Key words】** frame theory; frame network; microcomputer anti-disoperation system; logical judgment; network dimension; portability

DOI: 10.3969/j.issn.1000-3428.2013.12.005

### 1 概述

在线式微机防误系统是确保电力系统安全运行、防止人为误操作的重要设备, 倒闸操作经过该系统的模拟预演和逻辑判断, 能大大防止和减少电网事故的发生。随着电网的发展, 电力系统的可靠性要求越来越高, 在线式微机防误系统的作用也变得更为重要。但由于在线式微机防误系统存在可移植性不高的缺陷, 导致其在国内电力系统推广缓慢, 不利于电网的进一步发展<sup>[1]</sup>, 因此寻求提高微机防误系统可移植性的方法十分必要。

目前, 提高在线式微机防误系统可移植性主要有以下 3 种方法:

(1) 程序存储的方法<sup>[2]</sup>。其缺陷为: 要改动规则, 必须对原系统进行修改, 甚至要重新编写程序, 一般用户无法进行系统维护, 故其可移植性提高效果差。

(2) 基于规则推理的智能方法<sup>[3]</sup>。虽然该方法更加先进、智能, 可以提高在线式微机防误系统的可移植性, 但电力工作要求安全责任落实到人, 员工要亲自对系统进行每一步操作, 该方法完全依赖计算机操作而显得非常单一, 不符合电力系统的需求。

(3) “电路图+知识库+推理机”的方法<sup>[4]</sup>。由于大多数系统描述知识的效率较低, 而在推理过程中又要不断地访问知识库进行查询, 这将导致推理时间较长, 因此需要寻求一种高效的知识表示方法, 结合搜索算法, 就能使该形式得到最优化的运用, 提高在线式微机防误系统的可移植性。

本文以“电路图+知识库+推理机”方法为基础, 比较了逻辑表示法、语义网络表示法、框架表示法和面向对象表示法<sup>[5]</sup>等主流表示法, 最终利用框架表示法构建微机防误系统的设备与逻辑参数数据结构模型。

**作者简介:** 王友钊(1963—), 男, 副教授, 主研方向: 智能电网技术, 智能仪器; 黄冬, 硕士研究生

**收稿日期:** 2012-12-10 **修回日期:** 2013-01-07 **E-mail:** huangdong@zju.edu.cn

## 2 框架网络数据结构模型的描述、构建与优化

### 2.1 设备与逻辑参数信息的框架描述

框架(Frame)是将某类对象的所有知识组织在一起的一种通用数据结构, 一个框架由框架名和若干个槽(Slots)组成, 每个槽又由多个侧面(Facets)组成<sup>[6]</sup>。槽值描述框架所表示实体的各个属性, 侧面从不同方面来描述槽的特性。在较为复杂的框架中, 槽可以看成一种子框架, 子框架本身可以进一步分层次。框架之间的联系有纵向和横向 2 种方式。框架之间通过泛化关系(ISA 链、AKO 链或 PART-OF 链)连结成上下层的纵向联系<sup>[7]</sup>; 框架中槽值(同一框架中的 2 个槽值或不同框架之间的槽值)之间的关系形成了其横向联系。通过这 2 种联系方式, 各框架之间产生关联, 使整个框架系统构成一个具有层次的网络结构。

不同的框架网又可通过信息检索网络组成更大的框架网络, 代表一块完整的知识。框架网络的基本结构也是通过各框架之间的纵向或横向联系来实现的。框架网络在推理时由上下文中的数据选择候选框架, 通过上下文查询、默认、继承、附加等获得属性值, 按照框架中各个槽的次序填入, 使候选框架具体化, 以生成一个描述当前情况的实例。

一个复杂框架网络的各框架之间除了纵向联系和横向联系之外, 还会存在一些其他形式的联系。框架之间的联系是通过在槽中填入相应的框架名实现的, 它们之间的关系由槽名指定。在框架表示法中, 对于一些常用且可公用的槽名进行了标准化定义, 这些标准槽名被称为系统预定义槽名。下面给出一些常用的系统预定义槽名: ISA 槽, AKO 槽, Subclass 槽, Instance 槽, Part-of 槽, Infer 槽, Possible-Reason 槽, Similar 槽<sup>[8]</sup>。

当需要求解问题时, 首先把该问题用一个称为问题框架的框架表现出来, 然后将问题框架与知识库中的已有框架进行匹配, 找出一个或多个候选框架。当找到的候选框架多于一个时, 则通过某种筛选机制从中选出与问题框架最匹配的框架作为解框架。在找到解框架后, 框架的推理过程也就结束了。

在框架的层次型结构中, 上位框架为抽象度高的信息, 下位框架为具体的信息, 最下位框架对应于各个事例。框架结构的一般形式可表示为:

```

框架: 框架名, 框架类型
上位框架: 框架名
下位框架: 框架名
槽名 1: 侧面名 11(值 111, 值 112, ...)
侧面名 12(值 121, 值 122, ...)
.....
槽名 2: 侧面名 21(值 211, 值 212, ...)
侧面名 22(值 221, 值 222, ...)
.....

```

槽名 X: 侧面名 X1(值 X11, 值 X12, ...)

侧面名 X2(值 X21, 值 X22, ...)

.....

框架的槽和侧面可以被赋予各种类型的值, 既可以是数字、布尔值、字符串, 也可以是一个可执行的过程或动作, 甚至于是另一框架的框架名。槽和侧面还可以被赋予一个缺省值, 这样, 当系统或用户没有给槽或侧面指定值时, 框架将假设槽值或侧面值为缺省值。当给一个框架的各个槽和侧面赋予具体的值时, 就得到了该框架的一个实例框架。其中, 框架类型分为 4 种不同类型: Class: 抽象概念的框架; Instance: 具体示例的框架; Fact: 事实框架; Rule: 规则框架<sup>[9]</sup>。类似地, 可构建变电站的其他设备, 如断路器、网门等的框架。

#### 2.1.1 一次设备参数与信息的描述

以普通刀闸集为例, 该框架是在线式微机防误系统中的基础组成框架, 其余如断路器集等框架的组成与它大同小异。下面在例 1 中, 本文将介绍普通刀闸集参数模型框架表示的实现过程。

普通刀闸集的框架可表示为:

**例 1** 框架: <普通刀闸集>

上位框架: 一次设备图

下位框架: 刀闸 1, 刀闸 2, ..., 刀闸  $n$

框架: <刀闸 1>

ISA: 刀闸集

名称: ID(DZ1)

颜色: 设备颜色(红, 绿, 黄)

状态: 开合状态(0, 1)Default=0

大小: 设备大小(10, 10)

坐标: 左上角坐标(52, 18)

右下角坐标(42, 8)

普通刀闸集框架说明:

如上所示, 普通刀闸集框架的上位框架是一次设备图, 其下位框架为各个刀闸; 刀闸 1 框架中共有 6 个槽, 其中, ISA 槽是关系槽, 用来说明上下框架的联系, 刀闸 1 框架通过 ISA 这个关系槽与普通刀闸集框架建立关联, 而名称槽、颜色槽等是普通槽, 有说明槽属性的作用, 在状态槽中的开合状态侧面的缺省值为 0(断开)。坐标槽又有 2 个侧面: 左上角坐标和右下角坐标。

#### 2.1.2 逻辑知识库参数与信息的描述

以普通刀闸拉合条件集为例, 该框架是防误逻辑知识库中的基础组成框架, 其余如断路器拉合条件集等框架的组成与它大同小异。在例 2 中, 本文将介绍普通刀闸拉合条件集参数模型框架表示的实现过程。

**例 2** 在线式微机防误系统实现逻辑相关表示的步骤如下: 首先将写成 TXT 文件的防误逻辑程序读入系统, 随后在系统内部按照框架网络对各个条件进行表示。以刀闸 1 的合上条件为例, 其合上条件的逻辑程序为: DZ1 H:

DZ2=1, DLQ1=0, (DLQ2=0+DLQ3=0), WM1=!。其中, 普通刀闸拉合条件集的框架可表示为:

框架: <普通刀闸拉合条件集>

上位框架: 防误逻辑知识库

下位框架: 刀闸 1 拉合条件, 刀闸 2 拉合条件, ……  
刀闸 n 拉合条件

框架: <刀闸 1 拉合条件>

ISA: 普通刀闸拉合条件集

动作: (H, L)

运算符: (=, +, (, ), !)

名称: 设备名称(DZ2, DLQ1, DLQ2, DLQ3, WM1)

状态: (0, 1)

普通刀闸拉合条件集框架说明:

如上所示, 普通刀闸拉合条件集的上位框架是防误逻辑知识库, 其下位框架为各个刀闸拉合条件; 刀闸 1 拉合条件框架中共有 5 个槽, 其中, ISA 槽是关系槽, 用来说明上下框架的联系, 刀闸 1 拉合条件框架通过 ISA 这个关系槽与普通刀闸拉合条件集框架建立关联, 而动作槽、运算符槽等是普通槽, 有说明槽属性的作用。建立基础模型后, 组成了整个防误逻辑知识库参数模型的框架网络。

### 2.2 框架网络数据结构模型的构建

#### 2.2.1 一次设备电路图参数结构模型的构建

在线式微机防误系统中所有一次电气设备可以看作作为一个完整的知识空间, 它可以划分为普通刀闸、接地刀闸、断路器、网门、变压器等一些子空间, 它具有明显的层次型结构特点<sup>[10]</sup>。在使用框架表示法对设备信息进行描述后, 通过类似的一系列框架对在线式微机防误系统一次电气设备的描述, 可以建立起在线式微机防误系统一次设备电路图的框架网络表示模型, 图 1 为该模型的部分示意图。

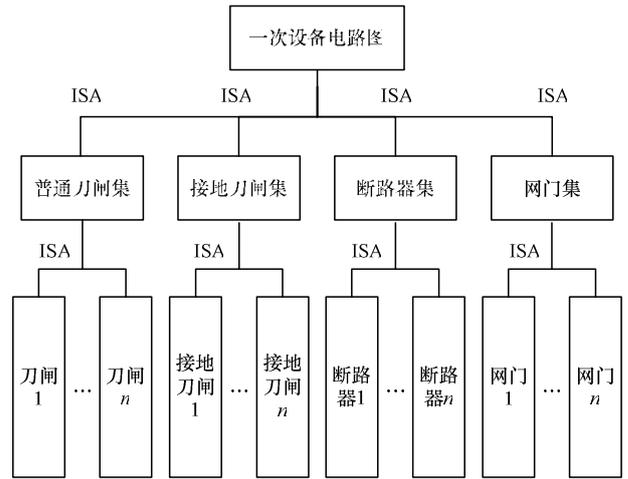


图 1 一次设备电路图参数结构部分示意图

#### 2.2.2 逻辑知识库参数结构模型的构建

与一次设备类似, 防误逻辑知识库也可以看作作为一个完整的知识空间, 它可以划分为普通刀闸拉合条件集、接地刀闸拉合条件集、断路器拉合条件集、网门拉合条件集等一些子空间, 同样具有明显的层次型结构特点。

在使用框架表示法对逻辑知识库信息进行描述后, 通过类似的一系列框架对逻辑知识库参数的描述, 可以建立起在线式微机防误系统逻辑知识库参数的框架网络模型, 与一次设备电路图的框架网络模型一起, 组成了在线式微机防误系统的整体框架网络结构模型, 成为实现“电路图+知识库+推理机”方法的基础。

在线式微机防误系统的整体框架网络结构模型如图 2 所示, 在接到点击设备操作的消息后, 推理机首先在生成的逻辑知识库框架网络中搜寻该设备拉合条件框架, 随后在一次设备电路图框架网络中寻取相对应基础框架的设备信息, 最后进行比对、做出判断。

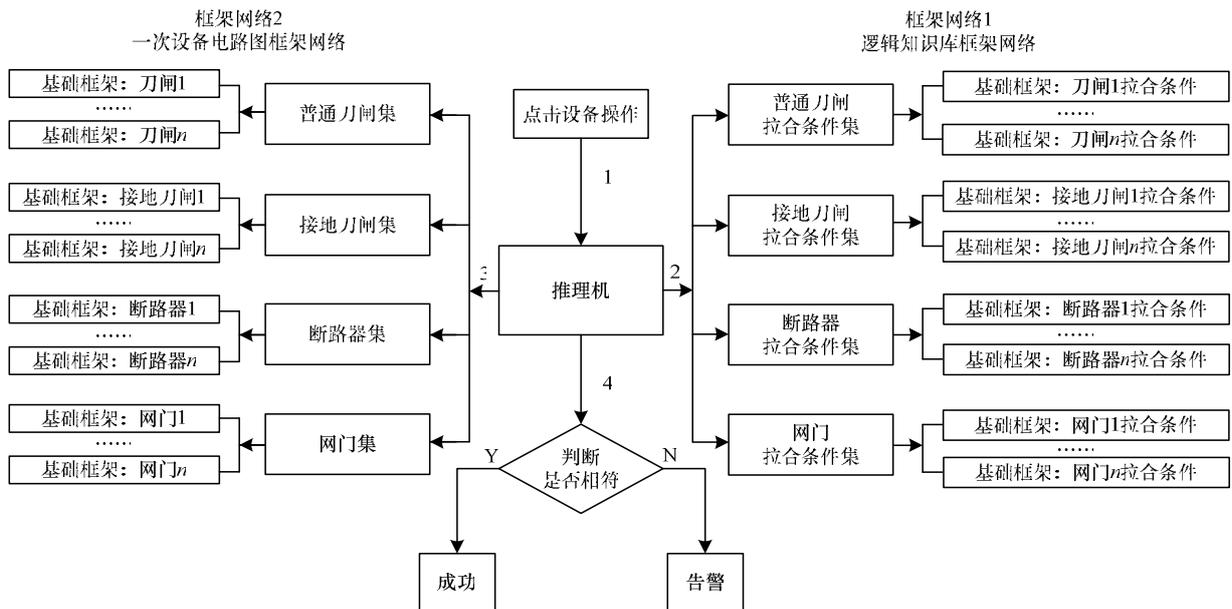


图 2 在线式微机防误系统的整体框架网络结构模型

### 2.2.3 框架网络数据结构的优化与改进

在建立了“一次设备电路图”框架网络数据结构和“逻辑知识库”框架网络数据结构的基础上,通过采用并结合 2 种方式对在线式微机防误系统的整体框架网络数据结构进行优化与改进,能够节省知识搜索时间,提高搜索效率,进一步提高在线式微机防误系统的可移植性。

(1)网络维度参数控制法。在系统整体框架网络中增加网络维度控制参数,系统先将写成 TXT 文件的防误逻辑程序读入系统,在生成“逻辑知识库”框架网络的同时,形成网络维度参数,自动控制系统在“一次设备电路图”框架网络结构中搜索知识的层次,与单纯地从首层到底层遍历“一次设备电路图”框架网络相比,减少对不必要层次的搜索,能够明显提高搜索效率。

(2)同步准备法。在读取“逻辑知识库”的逻辑程序、生成框架网络的同时,直接对“一次设备电路图”框架网络进行分层。减少在“一次设备电路图”框架网络中搜索知识的准备时间,从而加快系统的搜索速度。

2 种方法相结合的框架网络结构优化设计流程见图 3。

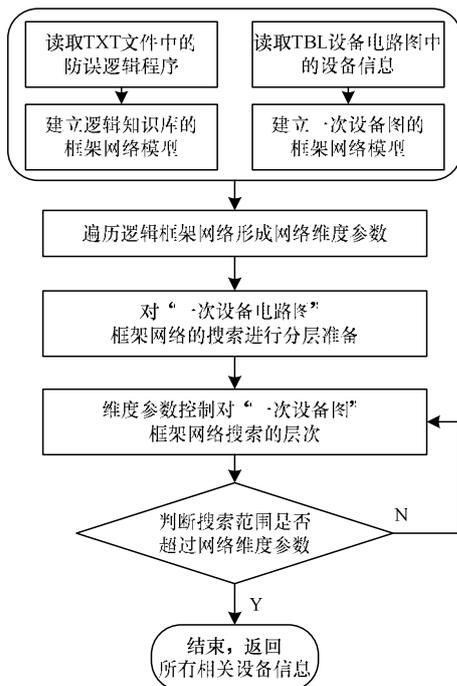


图 3 框架网络数据结构的优化与改进流程

### 2.3 基于 VC++的在线微机防误系统框架网络的实现

VC++作为一种面向对象的编程语言,其类的概念与框架极其相似。每一个类都有若干成员函数和成员变量,并可作为父类派生出若干子类,父类中的公有成员函数和成员变量可由子类继承,其类的结构也是层次型结构<sup>[1]</sup>。当用类对应框架、类的成员变量对应框架的槽、类的成员函数对应于框架中对槽的操作,就可建立类与框架间一一对应的关系,从而用 VC++就可很方便地实现用框架理论建立起来的数据模型。

本文在图形控件 Visual Graph(VG)的基础上,通过对在

线式微机防误系统中常见元件的基本属性以及它们在图元编辑状态的特性进行分析,抽象出它们的一些共有属性,如坐标、颜色、绘制、移动、旋转、删除、复制等,根据这些共有属性构造图元的基类,通过 VC++中的命令行“m\_vgMain.GetVg()”可以调用 VG 中相应设备的所需信息<sup>[12]</sup>。同理,通过 VC++的相关命令行可以实现逻辑程序的读取、框架网络的生成等功能。

不同层次之间存在继承关系,通过 VC++的赋值语句就可以在它们之间有选择地建立关联。由此可见,VC++中的父类与子类之间的派生与继承关系和框架中的上位框架与下位框架之间的纵向联系类似,同样在 VC++的类之间可以建立类似框架之间的横向联系。

本文利用框架理论构建的在线式微机防误系统元件参数的结构模型,运用面向对象的编程语言 VC++为编程工具,通过框架理论与 VC++的融合,采用有界深度优先搜索法与 BM 字符串匹配算法,就实现了一个具有较高可移植性和实用性的在线式微机防误系统。其中,基于框架理论构建的在线式微机防误系统的部分实现如图 4~图 6 所示。

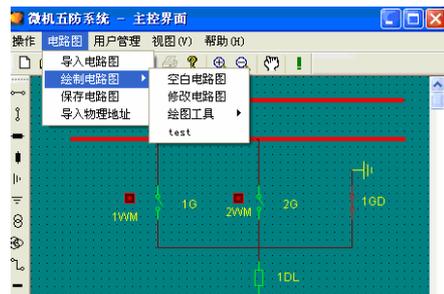


图 4 电路设备图的绘制、增加、清空和修改



图 5 电路图统一格式的整体移植

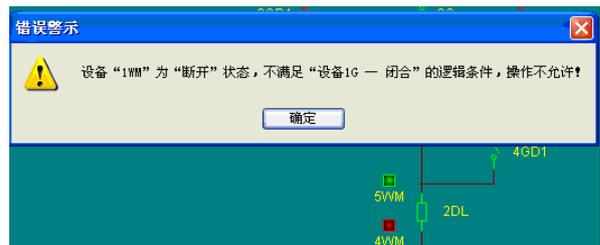


图 6 系统逻辑操作的判断

### 3 实验与结果分析

本文选取了语义网络表示法、面向对象表示法和逻辑表示法进行实现,对这 4 种表示法解决提高系统可移植性问题的程度与效率进行了分析与对比,以表格的形式给出了这 4 种方法在各比较项目中的结果。

本文通过使用专业的程序效率检验软件 EQATEC Profiler、EQATEC Tracer、Load Runner、PC-Lint、QAC 等

对原始框架网络表示法、优化后的框架网络表示法、语义网络表示法、面向对象表示法和逻辑表示法实现的在线式微机防误系统进行了比对,对其代码量、搜索时间、搜索准确率、可维护性(增加、删除、修改、全部重写)、占用磁盘空间、占用 CPU 和内存大小等性能,通过 20 次同硬件同任务测试取平均值的方式进行具体的比较,比较结果如表 1 所示<sup>[13-14]</sup>。

表 1 4 种表示法提高系统可移植性的效果对比

表示法	代码行数	搜索时间/s	搜索准确率/(%)	知识库的维护与移植				CPU 使用率/(%)	内存使用/KB
				增加内容	删除内容	修改内容	全部重写		
原始框架网络	2 300	4.2	99.3	快、方便	快、方便	快、方便	快、方便	0.79	54
优化框架网络	2 500	3.9	99.5	快、方便	快、方便	快、方便	快、方便	0.79	56
语义网络	4 200	6.9	85.9	慢、繁琐	慢、繁琐	慢、繁琐	慢、繁琐	1.25	230
面向对象	3 100	4.6	96.3	快、方便	快、方便	慢、繁琐	快、方便	0.62	69
逻辑表示	5 400	10.5	98.7	快、方便	快、方便	慢、方便	慢、繁琐	1.97	326

从表 1 的客观比较可以看出,针对提高在线式微机防误系统可移植性的问题,本文方法优化后的框架网络表示法在同种表示方法中最为适合,其搜索时间最短、搜索准确率最高,所占系统硬件资源少,能较大地提升知识在计算机中存储、检索、使用和修改的效率,对在线式微机防误系统的可移植性提高效果好。

### 4 结束语

本文提出一种基于框架理论的提高微机防误系统可移植性的方法,设计思路主要以框架理论为基础,将框架网络引入在线式微机防误系统,并通过网络维度参数控制法与同步准备法对框架进行自主优化;最后通过将框架理论融入面向对象的语言 VC++ 中,实现了基于框架网络数据结构的在线式微机防误系统软件。实验结果表明,本文方法提升了知识的检索和修改效率,节省了微机防误系统移植的时间,与其他主要表示方法相比,对于提高在线式微机防误系统可移植性的效果更好。今后将研究框架网络数据结构环境下的搜索算法,对其优化以提高该环境下算法的搜索速度与效率,从而缩短搜索时间,进一步提升系统的可移植性。

#### 参考文献

- [1] 谢 燕, 吕铁民. 微机防误闭锁系统的分析与探讨[J]. 信息技术, 2012, (2): 42-42.
- [2] 黄聪会, 陈 靖, 张 黎. 软件移植理论与技术研究[J]. 计算机应用研究, 2012, 29(6): 2024-2027.
- [3] 朱炳铨, 王 松, 黄晓明. 支持一键式顺控的智能操作票生成和管理系统[J]. 华东电力, 2011, 39(5): 837-841.
- [4] Hugo H, Danushka B, Mitsuru I. Semi-supervised Discourse Relation Classification with Structural Learning[C]//Proc. of CICLing Conference on Intelligent Text Processing and Computational Linguistics. [S. l.]: Springer, 2011.
- [5] 谢 锋. 基于框架理论的操作票自动生成专家系统[D]. 武汉: 武汉大学, 2003.
- [6] 耿汉杰. 110 kV 变电站操作票自动生成系统研发[D]. 武汉: 武汉大学, 2004.
- [7] 陈小红, 尹 斌, 金 芝. 基于问题框架的需求建模: 一种本体制导的方法[J]. 软件学报, 2011, 22(2): 177-194.
- [8] 刘迎春. 基于实时信息的电网调度操作票专家系统的设计与实现[D]. 武汉: 华中科技大学, 2007.
- [9] Xiang Tiejuan, Wu Hongqing, Xie Feng, et al. The Research and Realization of the Automatically Forming System of Operation Tickets[C]//Proc. of International Conference on Power System Technology. Kunming, China: [s. n.], 2002: 2140-2144.
- [10] 樊 鹏. 基于 GPS 的 SCADA-EMS 煤矿供电调度系统的研究[D]. 阜新: 辽宁工程技术大学, 2009.
- [11] 余南华, 黄 曙, 李先波. 变电站五防控制规则自动生成技术思路[J]. 电力系统自动化, 2010, 34(17): 97-99.
- [12] Li Cheng, Yuan Yubo, Luo Qiang. The Research and Application on Interfacing Technology Between Electronic Current Transformer and Relay Protection[C]//Proc. of 2011 International Conference on Advanced Power System Automation and Protection. Beijing, China: [s. n.], 2011.
- [13] 邵国庆, 张 豪, 邵 松. 数字化变电站的发展现状[J]. 工业设计, 2011, (6): 168-168.
- [14] Wang Longjun, Wang Gang, Li Bo. Impacts of Protection System Misoperation on the Reliability of Turkish National Power Transmission System[C]//Proc. of International Conference on Advanced Power System Automation and Protection. Beijing, China: [s. n.], 2011.

编辑 任青慧