

基于自动机模型的平面调车软件设计方法

魏 臻¹, 陆 阳¹, 汤 俊², 鲍红杰¹

(1. 合肥工业大学计算机与信息学院, 安徽合肥 230009; 2. 合肥工业大学高科信息技术有限公司, 安徽合肥 230088)

摘 要: 平面调车是一种铁路调车过程中的离散事件系统. 可靠性高、易测试的平面调车软件对保障铁路运输安全极为重要, 因此需要建立一种规范模型基础上的平面调车软件设计方法. 分析了平面调车系统中事件驱动过程的复杂性以及自然语言规则描述方式的缺陷, 提出通过自动机模型描述平面调车系统, 具体讨论了平面调车系统与时间自动机(TA)、下推自动机(PDA)和分层自动机的关系. 从状态划分、数据结构设计、状态转移流程等方面说明了基于自动机模型的平面调车软件的设计方法.

关键词: 离散事件系统; 平面调车; 时间自动机; 下推自动机

中图分类号: TP301, U283 **文献标识码:** A **文章编号:** 0372-2112 (2009) 05-1013-06

Design Method for Software of Flat Shunting Systems Based on Automaton Models

WEI Zhen¹, LU Yang¹, TANG Jun², BAO Hong jie¹

(1. College of Computer and Information, Hefei University of Technology, Hefei, Anhui 230009, China;

2. GOCOM Information & Technology Co. LTD, Hefei, Anhui 230088, China)

Abstract: Flat shunting is a type of discrete event systems in the process of railway dispatching. It is very important that software of any flat shunting system have high reliability and high testability for ensuring transportation safety. For that goal, it is necessary to build a kind of regular design methods for software of flat shunting systems. In this paper, the complexity of event driving in the process of flat shunting and the limitation of description with natural language are analyzed. Therefore new descriptions of some processes in a flat shunting system with automaton models are proposed. In detail, the relationships between a flat shunting system and timed automata (TA), pushdown automata (PDA) and the structure of layered automata are discussed. On the basis of explaining the partition of states, the data structure and the flow of state transition, a design method of flat shunting systems based on automaton models is built.

Key words: discrete event system; flat shunting; timed automaton; pushdown automaton

1 引言

在铁路运输作业过程中存在着大量的平面调车作业, 平面调车系统是典型的离散事件系统. 进行平面调车作业时机车司机根据调车员、连结员的信号对机车运行进行操控, 并配合连结员进行摘挂工作. 传统的平面调车作业是通过人工信号灯或信号旗的方式完成. 近年来随着通信技术和计算机技术的发展, 平面调车中广泛采用无线通信技术和嵌入式调车设备, 调车员、连结员将调车信号通过专用无线手持机发送到机车上, 由安装在机车上的嵌入式设备接收解码后根据调车信号的显示规则进行显示和语音提示, 指导机车司机进行机车运行的操控. 因此, 机车上嵌入式调车设备工作的可靠性

将直接影响到机车运行和地面作业人员的安全. 根据检索, 有关平面调车软件设计方法的文献十分少见, 分析原因可能是目前铁路行业对于平面调车需求说明主要依靠平面调车系统指令表^[1], 而实际上平面调车系统指令表并不能完全包含平面调车的需求, 对于这种情况各类开发人员在开发平面调车软件时只能根据自己的理解进行软件设计和编程, 缺乏系统的方法总结. 由于对系统的模型本质缺乏认识, 在平面调车软件开发时容易直观地导致采用规则式的方法, 具有一定的“经验”色彩, 不能充分保证软件的可靠性和可测试性, 而且在有些企业需要对平面调车系统指令表进行调整时(主要发生在企业自备铁路), 规则的修改也容易引起新的软件错误.

本文从自动机的角度研究平面调车信号的状态转移过程,提出了一种采用时间自动机模型设计平面调车软件的方法.这种方法具有严格的理论基础,能充分保证软件的设计质量,并且能为软件测试提供很大的便利,是一种规范的平面调车软件设计方法.

2 平面调车过程

平面调车的输出包括四个指示灯和语音提示,四个指示灯分别是:绿灯、黄灯、红1灯和红2灯.平面调车系统的结构见图1.

平面调车的过程描述主要依靠平面调车系统指令表,表1是某企业铁路执行的平面调车系统指令表,表中忽略了调车手持机发出命令时具体的按钮组合,而只以命令作为输入进行讨论.其中测机命令是调车员手持机向接收处理设备发送的通信联络信号,以确认当前无线信道是否通畅;链路故障和链路恢复是接收处理设备根据测机情况产生的内部事件,和调车命令一同作为系统的输入.

软件设计的实践说明,仅有平面调车系统指令表来描述平面调车过程是不充分的,在需求分析时需要进一步补充.经过充分的分析以及和用户不断的沟通,总结以下自然语言描述作为对表1的补充.

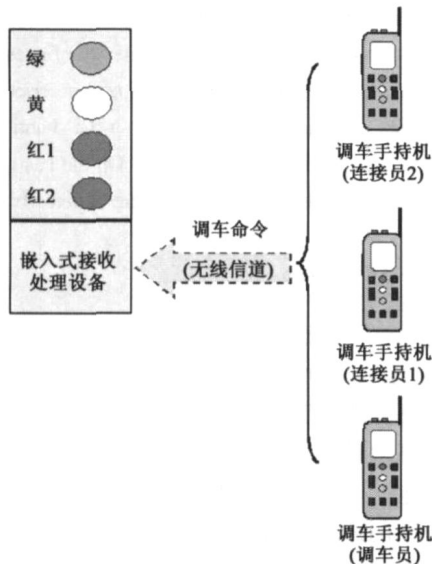


图1 平面调车系统结构

(1) 动车类命令包括:启动、推进、减速、十车、五车、三车、一车、稍行、反稍行;停车类命令包括:停车、紧急停车、连结;其它命令包括:鸣笛、拆除紧急停车、测机命令、链路故障、链路恢复.

(2) 动车状态下进行测机,停车状态下停止测机.

(3) 停车和紧急停车命令任何情况下均为有效命令;拆除紧急停车命令只在当前是“紧急停车”时有效;拆除紧急停车命令必须是谁紧急停车谁解锁,而且如

果收到两个连结员的紧急停车,必须两个连接员都拆除紧急停车方能实现解锁功能.

表1 平面调车系统指令表

命令/事件	命令/事件标识	灯显示	语音提示
停车	0	红1常亮	“停车”2次
牵引、起动	1	绿常亮	“启动”2次
推进	2	绿闪2次,每次0.5s	“推进”2次
减速	3	黄常亮	“减速”2次
十车	4	黄闪3次,每次0.5s	“十辆”2次
五车	5	黄闪2次,每次0.5s	“五辆”2次
三车	6	黄闪1次,每次0.5s	“三辆”2次
一车	7	黄闪1次,每次0.5s	“一辆”2次
连结	8	红1、绿交互3次,每次0.5s,停红1	“连结”2次
稍行	9	红1、绿交互3次,每次0.3s,停黄	“稍行”2次
反稍行	10	黄、绿交互3次,每次0.5s,停黄	“反稍行”2次
鸣笛	11	黄闪5次,每次0.5s	“鸣笛”2次
紧急停车1	12	红2常亮	“停车”2次
紧急停车2	13	红2常亮	“停车”2次
拆除紧急停车1	14	所有紧急停车命令都解锁后,红2灭	“解锁”2次
拆除紧急停车2	15	所有紧急停车命令都解锁后,红2灭	“解锁”2次
测机	16	重复执行推进	“推进”2次
链路故障(事件)	29	连续红闪,每次0.5s	“链路故障”2次
链路恢复(事件)	30	除红2以外,其余灯灭	“链路恢复”2次

(4) 紧急停车后任何动车命令都是无效命令.

(5) 停车命令后,紧急停车、启动、推进、鸣笛、稍行、反稍行、连接等命令是有效命令;减速、十车、五车、三车、一车、链路故障、链路恢复、拆除紧急停车、测机为无效命令.

(6) 红1、红2同时亮时,鸣笛命令、停车、紧急停车、拆除紧急停车是有效命令;启动、推进命令使红1灭并且没有语音输出;其它命令为无效命令.

(7) 启动、推进、减速命令互不相容.

(8) 十车、五车、三车、一车和鸣笛命令不改变当前红1、红2、绿灯的状态;如果执行前是黄灯常亮,执行完毕回到执行前的状态.

(9) 在动车状态下十车、五车、三车、一车命令为有效命令.

(10) 在稍行、反稍行和连结命令执行的过程中,停车、紧急停车、链路故障命令为有效命令,其它命令为无效命令;在执行结束后稍行和反稍行命令作为减速命令处理,连接命令作为停车命令处理.

(11) 在动车状态下 T_{link} , 时间内未收到测机信号判为链路故障;在链路故障情况下,收到测机、启动、推进命令后判为链路恢复.

(12) 鸣笛命令在稍行、反稍行和连接命令的执行过程中无效, 其它情况下均为有效; 鸣笛命令与停车、启动、推进、链路故障和紧急停车命令相容, 与减速、十车、五车、三车、一车命令不相容。

(13) 在推进状态下每收到一个测机命令重复执行一次推进命令, 在其它动车状态下测机命令无效。

(14) 在无灯的情况下所有动车类命令均为有效命令。

仔细分析可以发现, 以上 14 条自然语言的补充规则存在语义重叠现象, 而且因为缺乏理论的依据, 并不能证明补充的这些自然语言条件就一定能够完备描述平面调车过程, 而且自然语言还需要能够转换成计算机系统可以识别的规则表达结构才能由软件编程实现。在现实情况下, 各个企业平面调车需求的不一致性还会导致自然语言条件的经常修改。所以, 平面调车过程实际上是一个比较复杂的离散事件驱动系统, 只有建立在适当的理论模型分析的基础上才能保证软件的可靠性、可测试性和可重用性。

3 平面调车过程的自动机模型

平面调车过程涉及的自动机模型包括: 时间自动机(TA: Timed Automaton)、下推自动机(PDA: Pushdown Automaton)和分层自动机结构。下面对平面调车和这些自动机模型的关系逐一进行分析。

3.1 时间自动机模型

时间自动机是由 Alur 于 1994 年提出的^[2], 对于分析实时事件系统有重要的意义^[3-5]。

时间字是由字母表 Σ 上的无限字符串 $\sigma = \sigma_1 \sigma_2 \dots$ 和时间域上的时间序列 $\tau = \tau_1 \tau_2 \dots$ 构成的一个序偶 (σ, τ) , 时间语言是时间字的集合, 时间自动机接受时间语言。

定义 1 X 是时钟变量集, 时间约束 δ 的集合 $\Phi(X)$ 递归地定义如下:

$$\delta := x \leq c \mid c \leq x \mid \neg \delta_1 \wedge \delta_2$$

$x \in X, c$ 是有理数常量。

时钟约束表示某一个特定事件发生时时钟应满足的条件。

一个时钟集 X 的时钟解释 v 表示为 X 的每一个时钟赋一个实数值, 也就是从 X 到实数集 R 的一个映射。称 X 的一个时钟解释满足时钟约束 δ 当且仅当 v 的值使 δ 为真。设 $t \in R, v + t$ 表示对 x 赋值 $v(x) + t; v \cdot t$ 表示对 x 赋值 $v(x) \cdot t$; 对于 $Y \subseteq X, [Y \rightarrow t]v$ 表示 X 的时钟解释, 对任意的 $x \in Y$, 赋值 t , 而其余的时钟满足 v 。

定义 2 一个时间自动机是一个多元组 $A = (\Sigma, S, S_0, C, T, F)$ 。 Σ 是字母表, S 是有限状态集合, $S_0 \subseteq S$ 是初始状态集, C 是有限时钟集, $T \subseteq S \times S \times \Sigma \times 2^C \times$

$\Phi(C)$ 是状态转移函数, F 是接受条件, A 在时间字 (σ, τ) 上的一个运行 r 是形如

$$r: \langle s_0, v_0 \rangle \xrightarrow{\sigma_1 / \tau_1} \langle s_1, v_1 \rangle \xrightarrow{\sigma_2 / \tau_2} \langle s_2, v_2 \rangle \xrightarrow{\sigma_3 / \tau_3} \dots$$

的无序序列: $s_i \in S, v_i \in \{C \rightarrow R\}, i = 0, 1, \dots$, 使得 $s_0 \in S_0, v_0(x) = 0, x \in C$, 对任意的 $i \geq 1$ 有 $\langle s_{i-1}, s_i, \sigma_i, \lambda_i, \xi_i \rangle \in T$ 使 $(v_{i-1} + \tau_i - \tau_{i-1})$ 满足 ξ_i , 且 $v_i = [\lambda_i^* 0] (v_{i-1} + \tau_i - \tau_{i-1})$ 。

设平面调车的命令/事件集和空字符 ε 构成字母表 Σ_{TA} , 平面调车的一个运行由 $\sigma =$ 推进 减速 一车 连结 推进 测机 ... 和时间序列 τ 组成, 一次灯的闪灭周期为 1s, 则根据表 1 和相关的自然语言描述可以构造如图 2 的时间自动机, 其中 x 是时间变量。

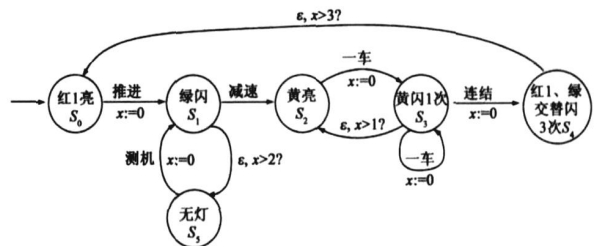


图2 平面调车时间自动机范例

图 2 中状态 S_1 不接收测机命令, 而状态 S_3 接收测机命令; 状态 S_4 不接收推进命令, 而状态 S_0 接收推进命令。

实际的平面调车过程远比图 2 复杂, 但不论怎样复杂, 平面调车过程中的大部分状态转移过程都离不开时间的因素, 因此在模型上都和图 2 类似。

第 1 节中的第(2)、(11)条关于链路判断的描述也是一个时间自动机, 该自动机具有事件输出 E , 采用广义自动机模型, 如图 3 所示, 其中输入字符表 $\Sigma_{input} = \{\varepsilon, \text{调车命令} \mid \text{命令/事件标识} = 0 \sim 16\}$, y 为时间变量, E 值对应表 1 中相应的“命令/事件标识”值。

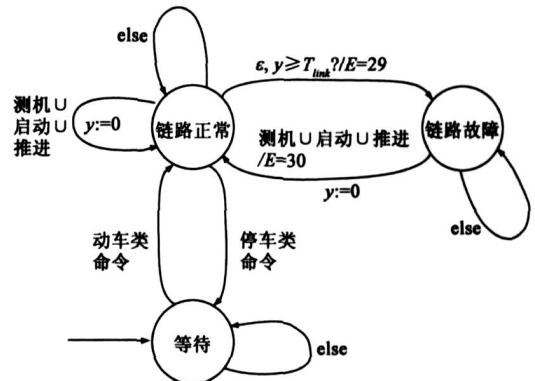


图3 链路判断自动机

3.2 与下推自动机的关系

下推自动机在有限自动机的基础上增加了一个无限深度的栈, 栈顶内容和状态转移函数关联, 并且每一

次状态转移时可以调整栈顶内容^[6,7].

定义 3 一个下推自动机是一个多元组 $P = (\Sigma, S, S_0, T, \Gamma, F)$. Σ 是字母表, S 是有限状态集合, $S_0 \subseteq S$ 是初始状态集, $T \subseteq S \times S \times \Sigma \times \Gamma$ 是状态转移函数, Γ 是栈字母表, F 是接受条件.

下推自动机的状态转移形式是 $(S_x, aw, Z_x\beta) \vdash (S_y, w, Z_y\beta)$, 表示消耗输入 a , 并且用 Z_y 来替代栈顶的 Z_x 后, 状态从 S_x 转移到 S_y . 在自动机状态图中可简化表示为 $(a, Z_x \rightarrow Z_y)$. 特别地, $(a, \varepsilon \rightarrow \varepsilon)$ 表示消耗字符 a , Z_y 压入栈顶; $(a, Z_x \rightarrow \varepsilon)$ 表示消耗字符 a , Z_x 弹出栈顶; $(\varepsilon, Z_x \rightarrow Z_y)$ 表示不消耗任何输入, Z_y 替代栈顶的 Z_x .

第 1 节中第(3)条关于紧急停车和拆除紧急停车的自然语言描述可以通过一个下推自动机进行形式化描述. 在这个下推自动机中, 设 $\Sigma_{PDA} = \{\varepsilon, a_1, a_2, b_1, b_2\}$, 其中 $a_1 =$ 紧急停车 1, $a_2 =$ 紧急停车 2, $b_1 =$ 拆除紧急停车 1, $b_2 =$ 拆除紧急停车 2; $\Gamma = \{\$, Z_1, Z_2, Z_3\}$, 其中 $\$$ 为栈底字符, 对应的状态图如图 4.

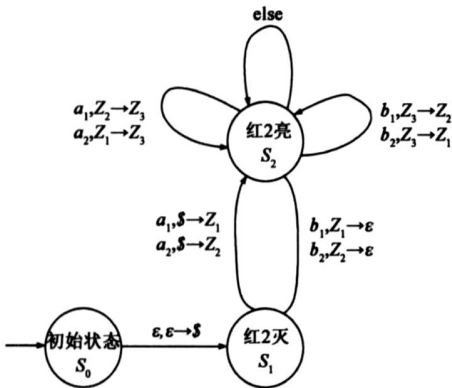


图 4 紧急停车及拆除过程的下推自动机

图 4 中的“else”弧表示在 S_2 状态下其他未列出的状态转移条件不引发状态转换和栈的变化

如果将图 4 的过程抽象成形式语言, 则图 4 识别形式语言 $L = \{L_1^* L_2^* \mid p \geq \min\{n, 1\}, q \geq \min\{m, 1\}\}$, 其中 $L_1 = \{a_1^n + a_2^m \mid n, m \geq 0\}$, $L_2 = \{b_1^p + b_2^q \mid p, q \geq 0\}$.

平面调车中的紧急停车和拆除过程其实也可以用有限自动机来表达的, 但需要对“红 2 亮”状态进行拆分, 变成四个不同的“红 2 亮”状态. 从问题的一般性考虑, 不妨假设一下, 如果有 n 个连结员, 那么“红 2 亮”需要拆分为 2^n 个状态, 有限自动机会变得很庞大; 此时用下推自动机表达的状态图仍然比较简单, 但下推自动机的转移函数关系也会变得复杂, 不过通过适当地修改转移规则形式, 可以化简这种复杂性.

3.3 分层的自动机结构

如果一个自动机带有输出, 并且该输出是另外一个自动机的输入, 就构成了分层的有限状态自动机. 在

通常的情况下, 分层的状态机都可以纳入 CFSM 模型中 (Communicating Finite State Machine)^[8,9].

平面调车过程的链路自动机和调车自动机之间就构成了分层自动机, 如图 5 所示.



图 5 分层的自动机结构

4 软件设计

在自动机软件的设计中, 可重用的实现框架是开发者最关注的内容之一^[10,11], 时间自动机软件也应遵循这一原则, 同时还要考虑时间变量对软件处理结构的影响. 根据前文分析的自动机模型, 平面调车的软件设计重点主要体现在状态划分、数据结构设计、状态转移流程设计等几个方面.

4.1 状态划分

平面调车自动机的状态空间设计比较适合以灯的状态为基础, 四个灯存在闪动状态和不同的组合, 有些组合是合法的, 有些组合是不合法的, 需要在状态设计时予以区分. 根据分析, 对于任何的平面调车系统指令表, 都可以将灯状态划分为基本灯状态和组合灯状态. 所谓基本灯状态是指每一个灯的常亮状态、闪动状态、多个灯交互闪动状态; 组合状态是基本状态的合法组合, 基本状态和组合状态的集合构成了自动机的状态空间. 表 2 是根据表 1 进行的状态划分.

表 2 状态划分

基本状态	A(绿常亮), B(黄常亮), C(红 1 常亮), D(红 2 常亮), E(绿闪 2 次), F(黄闪 5 次), G(黄闪 3 次), H(黄闪 2 次), I(黄闪 1 次), J(红 1、绿互闪 3 次), K(红 1、黄互闪 3 次), L(黄、绿互闪 3 次), M(红 1 闪), P(无灯)
组合状态	AF, AG, AH, AI, EF, EG, EH, EI, CD, CF, DF, CDF, MD, MF, MDF

4.2 数据结构设计

在平面调车的时间自动机中, 有些状态在没有输入的情况下, 经历一段时间后会自动转移到另一个状态, 而另一些状态在没有输入的情况下却始终稳定的. 所以在设计数据结构时要能够将这两种不同情况统一起来, 并考虑局部下推自动机和分层结构的数据结构要求. 此外, 由于状态的转移过程易于用查表方法实现, 数据结构的设计还要兼顾查表的便利性. 由于状态转移的自变量不再只是状态 ID 和命令/事件标识, 还会包括时间变量值和下推自动机的栈顶内容, 而在计算机环境下, 表格的建立不能引入连续的时间值作为自变量, 所以时间变量的处理必须由程序完成. 而下推自动机的状态转移表也应单独建立.

对自动机状态设立属性集 = { 状态 ID、状态执行、语音执行、下推自动机标志 }，数据结构如下：

```
typedef struct{
    unsigned short StateID;
    StateExecuteType StateExecute;
    VoiceExecuteType VoiceExecute;
    unsigned short PDAFlag; // 表明是否属于下推自动机的状态
}StateAttributeType
```

其中状态执行和语音执行是结构属性，语音执行结构为每个状态下播放语音的段号和播放次数，状态执行结构用于规定灯的操作方式，包括常亮、(多次) 闪亮、熄灭以及闪烁频率等，具体如下：

```
typedef struct{
    unsigned short L1_ OnTime; // 灯 1 点亮时刻
    unsigned short L1_ OffTime; // 灯 1 熄灭时刻
    unsigned short L1_ CtrlPerd; // 灯 1 控制周期
    unsigned short L1_ CtrlTimer; // 灯 1 控制计时器
    unsigned short L1_ CycNumb; // 灯 1 循环控制次数
    .....
    unsigned short L4_ OnTime; // 灯 4 点亮时刻
    unsigned short L4_ OffTime; // 灯 4 熄灭时刻
    unsigned short L4_ CtrlPerd; // 灯 4 控制周期
    unsigned short L4_ CtrlTimer; // 灯 4 控制计时器
    unsigned short L4_ CycNumb; // 灯 4 循环控制次数
} StateExecuteType
```

除了以上数据结构之外，还需设立 TA 暂态转换表 (TA_T_Tab)、TA 稳态转换表 (TA_S_Tab)、PDA 状态转移表 (PDA_Tab) 和时间复位表 (TimeRes_Tab)。TA_T_Tab 用于命令/ 事件发生作用引起的状态转移，TA_S_Tab 用于无命令/ 事件作用时由时间变量引发的状态转移，由于此时时间变量不宜作为查表自变量，因此必需借助暂态中的 StateID 作为自变量，PDA_Tab 只在图 4 的状态下发生作用，另外链路自动机依据类似的方法单独设计。

4.3 状态转移流程

时间自动机的状态转移条件可以划分为以下几种：

- (1) 输入字符引发状态转移，无时间约束条件；
- (2) 输入字符附加时间约束条件引发状态转移；
- (3) 无输入字符，由时间变量根据时间约束条件引发状态转移；
- (4) 相同输入字符附加不同时间约束引发不同的状态转移。

在针对平面调车软件的研究中，设计了图 6 的状态转移流程，适用于前三种情况。第(4)种情况比较特别

涉及时间约束的离散化表达问题和时间自动机可达性问题，在查表方法设计上比较复杂，相关的软件设计方法需要继续研究。

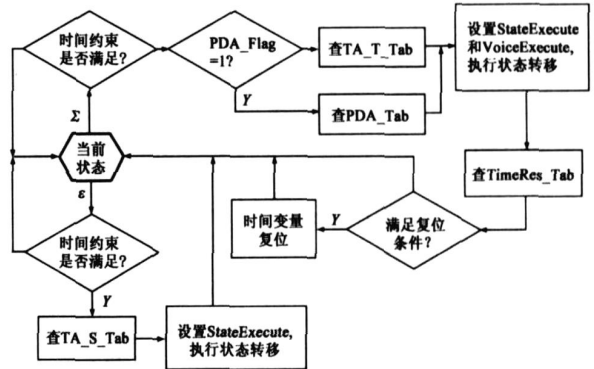


图6 状态转移流程

5 结论

工业控制过程中存在大量事件触发系统，有限自动机理论是分析它们的有力工具，尤其是时间自动机模型，在事件触发的实时系统建模方面有广泛的价值。平面调车的过程是一个复杂的事件触发系统，建立规范的软件设计方法对此类软件的开发有重要的指导意义。实践经验说明，仅仅采用自然语言的规则描述方法，很难构造一个可靠性好，易测试，易维护，适用需求更改的平面调车软件。本文采用的基于自动机模型设计方法可以适用各类平面调车软件开发，软件结构统一，主要处理部分可重用，在某大型钢铁企业的平面调车系统中获得成功应用，对其它具有同类特点的事件触发实时系统的建模分析也有借鉴意义。

参考文献：

- [1] TB/T 2834 2002, 铁路平面无线调车设备技术条件[S] .
- [2] Alur R, Dill D L. A theory of timed automata[J]. Theoretical Computer Science, 1994, 126: 183- 225.
- [3] Kaynar D K, Lynch N, Segala R, Vaandrager F. Timed I/O automata: A mathematical framework for modeling and analyzing real-time systems[A]. Proc of the 24th IEEE International Real-Time Systems Symposium [C]. Cancun, Mexico, IEEE Press, 2003. 166- 177.
- [4] 赵琛. 面向媒体时序描述的带时间自动机的自动构造方法[J]. 计算机学报, 1999, 22(12): 1289- 1294. Zhao Chen. Construction of timed automata for temporal specification of a distributed multimedia software system[J]. Chinese Journal of Computers, 1999, 22(12): 1289- 1294. (in Chinese)
- [5] 宋煌, 庄雷, 苏锦祥, 周清雷. 一种改进的区域自动机构造方法[J]. 计算机研究与发展, 2002, 39(3): 607- 611. Song Huang, Zhuang Lei, Su Jinxiang, Zhou Qinglei. An improvement on the construction of region automaton[J]. Journal

- of Computer Research and Development, 2002, 39(3): 607-611. (in Chinese)
- [6] Hopcroft J E, Motwani R, Ullman J R. 自动机理论、语言和计算导论[M]. 北京: 机械工业出版社, 2004.
Hopcroft J E, Motwani R, Ullman J R. Introduction to Automata Theory, Languages, and Computation [M]. Beijing: China Machine Press, 2004. (in Chinese)
- [7] Sipser M. 计算理论导引[M]. 北京: 机械工业出版社, 2000.
Sipser M. Introduction to the Theory of Computation[M]. Beijing: China Machine Press, 2000. (in Chinese)
- [8] Miller R E. Passive testing of networks using a CFSM specification [A]. Performance, Computing and Communications, 1998. IPCCC' 98[C]. IEEE International, Tempe/Phoenix, AZ, IEEE Press, 1998. 111- 116.
- [9] Peng W, Makki K. Reachability and reverse reachability analysis of CFSMs [J]. Computer Communication, 1996, 19: 668-674.
- [10] 敖丽敏, 王树大, 郭永洪. 一种基于有限自动机的软件工作流开发模型[J]. 计算机工程与应用, 2004, 40(28): 208-211.
Ao Limin, Wang Shuda, Guo Yonghong. A developing model for software workflow based finite automation [J]. Computer Engineering and Applications, 2004, 40(28): 208-211. (in Chinese)

- [11] 徐小良, 汪乐宇, 周泓. 有限状态机的一种实现框架[J]. 工程设计学报, 2003, 10(5): 251- 255.
Xu Xiaoliang, Wang Leyu, Zhou Hong. Implementation framework of finite state machines [J]. Journal of Engineering Design, 2003, 10(5): 251- 255. (in Chinese)

作者简介:



魏 臻 男, 1965 年生于安徽无为县, 合肥工业大学教授, 博士生导师, 主要研究方向为计算机控制、可靠性工程和铁路信号。
E mail: weizhen@gocom. cn



陆 阳 男, 1967 年生于安徽合肥市, 合肥工业大学教授, 博士生导师, 主要研究方向为计算机控制、嵌入式系统。
E mail: luyang_hf@126. com