



(12)发明专利

(10)授权公告号 CN 106407579 B

(45)授权公告日 2019.06.04

(21)申请号 201610854959.0

(22)申请日 2016.09.27

(65)同一申请的已公布的文献号
申请公布号 CN 106407579 A

(43)申请公布日 2017.02.15

(73)专利权人 南京国电南自轨道交通工程有限
公司

地址 210032 江苏省南京市新模范马路38
号

(72)发明人 李佑文 褚红健 罗存 曹黎明
王志心

(74)专利代理机构 南京知识律师事务所 32207
代理人 张苏沛

(51)Int.Cl.
G06F 17/50(2006.01)

(56)对比文件

CN 101546472 A,2009.09.30,

CN 105159178 A,2015.12.16,

CN 105718686 A,2016.06.29,

黄成玉等.新型矿井供电自动化远程监测
监控系统的设计与实现.《电气应用》.2012,第31
卷(第4期),第84-88页.

审查员 陈鸣

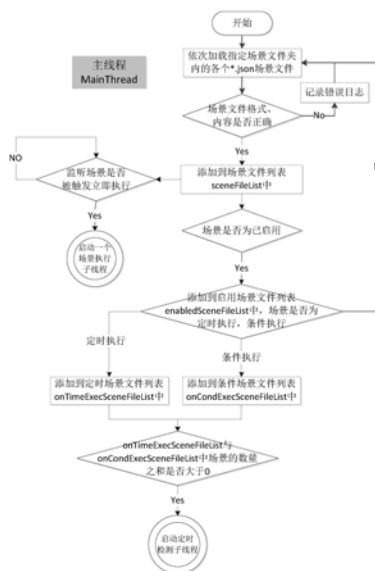
权利要求书1页 说明书6页 附图3页

(54)发明名称

一种电力监控仿真系统中场景设计方法

(57)摘要

本发明公开了一种电力监控仿真系统中场景设计方法,包括:基于数据点对象的场景配置文件结构设计,使用配置文件的形式来组织场景中出现的各种数据,每个配置文件中包含了场景的基本信息、场景步骤、以及场景步骤所使用的变量信息,场景配置文件结构使用JSON数据格式进行组织;基于多线程的后台场景实现方式,包括三种线程,主线程1个,定时检测子线程1个,场景执行子线程N个(0≤N视同时执行的场景个数而定)。本发明在电力监控仿真系统中引入场景设计的方法,场景执行使用多线程并行的方式极大地优化了事故仿真的流程和事故场景的并行性,提高了仿真效率。



1. 一种电力监控仿真系统中场景设计方法,其特征在于,包括:

基于数据点对象的场景配置文件结构设计,使用配置文件的形式来组织场景中出现的各种数据,每个配置文件中包含了场景的基本信息、场景步骤以及场景步骤所使用的变量信息,场景配置文件结构使用JSON数据格式进行组织;

基于多线程的后台场景实现方式,包括三种线程,主线程1个,定时检测子线程1个,场景执行子线程N个, $0 \leq N$;

主线程是场景执行进程的后台主线程,其负责场景文件的解析与加载,监听所有场景文件是否需要立即执行,并负责启动定时检测子线程;

定时检测子线程由主线程启动,其每隔1秒钟对定时类型的场景进行检测,定时时间到启动子线程执行该定时场景,同时对条件类型的场景条件表达式进行检测,若其条件表达式的值从false变为true则启动子线程执行该条件场景,需要注意的是对于循环类型的条件执行场景,并不会在条件为true时一直循环执行场景内容,而是每次条件从false变为true时执行一次,从而达到仿真实际故障发生时出现相应状态的效果;

场景执行子线程负责场景内容的具体执行,其由主线程、定时检测子线程启动,各个场景执行子线程之间相互不受影响。

2. 根据权利要求1所述的电力监控仿真系统中场景设计方法,其特征在于:

所述数据点对象,包括遥信点、遥测点、遥控点和遥脉点,每个点对象又包括了多个属性,每个属性对应一个值;

所述场景步骤,每个场景步骤包含一个用于唯一标识该场景步骤的alias、数据点对象的ObjectId、数值类型属性attribute、数值类型value,以及一个event事件组合,event中包含有用于表示在执行这个步骤前延时的毫秒时间delay、用于表示执行这个步骤的前提条件condition。

3. 根据权利要求1所述的电力监控仿真系统中场景设计方法,其特征在于:场景指的是特定条件、事件发生时导致的一系列顺序变化的数据集合。

4. 根据权利要求1所述的电力监控仿真系统中场景设计方法,其特征在于,触发场景的执行方式一共有三种:

定时执行:要求场景配置为启用,定时时间到后,立即执行场景,若场景为循环场景则一直在后台循环执行场景步骤;

条件执行:要求场景配置为启用,后台循环检测场景的条件表达式,在每次条件表达式的结果从false变为true时执行一次场景;

立即执行:不要求场景配置为启用,仿真HMI界面触发场景立即执行后直接将场景执行一次。

5. 根据权利要求4所述的电力监控仿真系统中场景设计方法,其特征在于:

所有场景的执行都受控于开始、暂停、结束三种状态,只有开始状态时才能执行,当为暂停状态时所有已经在执行中的场景会在执行完当前所执行的这一步骤之后暂停后面的执行步骤,等待再变为开始状态后继续执行,当为结束状态时结束所有场景的执行。

一种电力监控仿真系统中场景设计方法

技术领域

[0001] 电力监控系统(简称PSCADA:Power Supervisory Control and Data Acquisition)是用于监视和控制电力生产和供应过程的、基于计算机及网络技术的业务系统及智能设备,以及作为基础支撑的通信及数据网络等,它是构建众多供电、配网自动化应用系统的基础。本发明涉及城市轨道交通电力监控系统、电力监控仿真系统、电力监控培训系统以及相关电力调度培训系统等领域。

背景技术

[0002] 随着世界经济、人口的快速发展,人们对于方便、快捷、舒适的交通运输有着越来越高的要求、在水陆空飞速发展的同时,城市轨道交通也得到了飞速的发展,尤其是以地铁、城轨、轻轨为代表的城市轨道交通,在迅速发展的同时,高度安全可靠、经济合理适用、调度操作方便的电力供给是城市轨道交通安全、正常、有效运营的重要条件和保证。牵引供电系统是城市轨道交通供电系统的重要组成部分,直流牵引系统的正常运行、为电力机车提供电能,是整个供电系统的最终目标。车辆在直流牵引系统的作用下,沿铁轨和接触网运行,实现交通功能。牵引供电系统的特殊使命要求其运行安全必须得到保障,对牵引供电系统进行远程状态监测和常规控制是非常重要的保障措施之一。

[0003] 随着计算机技术、通讯技术和人工智能技术的快速发展,电力监控系统在电力行业及其他相关行业得到了越来越广泛的应用,城市轨道交通电力监控培训仿真系统的重要性显得更加重要,国内各大厂家都开始着力于该类仿真系统的研发,因此电力监控仿真培训系统的功能经历了从单一到综合,从简单到复杂的发展过程,其作用也由演示性转变为指导性,并成为电力监控自动化的一个重要产品和控制中心培训操作员的得力工具。

[0004] 城市轨道交通电力监控系统对轨道交通供电系统主变电所、牵引变电所、降压变电所等全线供电设备数据的监视和控制,以确保牵引供电和全线电力变配电系统安全可靠和经济运行,其主要实现遥信、遥测、遥控、遥脉功能。具体包括实现对各种设备的控制、信息采集、数据分析处理、事故报警、画面调阅、数据查询等功能。

[0005] 由于城市轨道交通电力监控系统的复杂性和系统设备价格昂贵,以及受到现场环境的应用、安全性能等限制,往往不能直接在现场进行系统的学习和培训,因此有必要设计城市轨道交通电力仿真培训系统,其主要目的就是打破现场学习的各种限制,采用数字化的仿真培训系统,不仅能够大幅减少运营公司的成本,提高现场运行的安全性,而且提前为员工进行培训,并大幅提高培训的效果和进度,增强社会效益。

[0006] 城市轨道交通电力监控仿真培训系统是对真实地铁电力运行情况的仿真模拟,其设计的目的是为了能够实现一种能够对地铁电调人员操纵电力设备和处理故障情况的培训。该系统的一个主要功能就是能够仿真模拟轨道交通电力监控系统在实际中经常遇到的各种故障场景(如接触网短路、交流侧短路、一个主所故障、框架保护动作场景等等)、操作场景(如区间紧急停电、双边供电、大双边供电、单边供电场景等等)。

发明内容

[0007] 针对现有技术中存在的问题,基于各种场景在仿真系统中的重要性,本发明提出了一种具有场景共性的场景设计与实现方案。

[0008] 本发明的技术方案为:一种电力监控仿真系统中场景设计方法,包括:

[0009] 基于数据点对象的场景配置文件结构设计;

[0010] 设计良好的数据组织方式可以为场景过程的配置、场景过程的后台执行提供更好的支持,而且数据组织作为场景功能的基本接口,其连接了场景的前台配置和后台执行两个过程,二者通过这种组织方式的配置文件进行交互,而且数据组织其逻辑的设计合理性也增强了场景功能的可扩展性。

[0011] 使用配置文件的形式来组织场景中出现的各种数据,每个配置文件中包含了场景的基本信息、场景步骤、以及场景步骤所使用的变量信息,场景配置文件结构使用JSON数据格式进行组织。

[0012] 数据点对象,包括遥信点、遥测点、遥控点和遥脉点,每个点对象又包括了多个属性,每个属性对应一个值。

[0013] 场景步骤,每个场景步骤包含一个alias(用于唯一标识该场景步骤)、id(数据点对象的ObjectId)、attribute(数值类型属性)、value,以及一个event事件组合,event中包含有delay(用于表示在执行这个步骤前延时的毫秒时间)、condition(用于表示执行这个步骤的前提条件)。

[0014] 基于多线程的后台场景实现方式;

[0015] 触发场景的执行方式一共有三种:

[0016] 定时执行:要求场景配置为启用,定时时间到后,立即执行场景,若场景为循环场景则一直在后台循环执行场景步骤;

[0017] 条件执行:要求场景配置为启用,后台循环检测场景的条件表达式,在每次条件表达式的结果从false变为true时执行一次场景。

[0018] 立即执行:不要求场景配置为启用,仿真HMI界面触发场景立即执行后直接将场景执行一次。

[0019] 所有场景的执行都受控于开始、暂停、结束三种状态,只有开始状态时才能执行,当为暂停状态时所有已经在执行中的场景会在执行完当前所执行的这一步骤之后暂停后面的执行步骤,等待再变为开始状态后继续执行,当为结束状态时结束所有场景的执行。基于多线程的后台场景实现方式中包括三种线程,主线程1个,定时检测子线程1个,场景执行子线程N个($0 \leq N$ 视同时执行的场景个数而定)。

[0020] 主线程是场景执行进程的后台主线程,其负责场景文件的解析与加载,监听所有场景文件是否需要立即执行,并负责启动定时检测子线程;

[0021] 定时检测子线程由主线程启动,其每隔1秒钟对定时类型的场景进行检测,定时时间到启动子线程执行该定时场景,同时对条件类型的场景条件表达式进行检测,若其条件表达式的值从false变为true则启动子线程执行该条件场景,需要注意的是对于循环类型的条件执行场景,并不会在条件为true时一直循环执行场景内容,而是每次条件从false变为true时执行一次,从而达到仿真实际故障发生时出现相应状态的效果。

[0022] 场景执行子线程负责场景内容的具体执行,其由主线程、定时检测子线程启动,各

个场景执行子线程之间相互不受影响。

[0023] 本发明的有益效果：

[0024] 本发明在电力监控仿真系统中引入场景设计的方法，简化了在仿真系统中遇到的各种故障和操作流程，将所有故障、操作导致的一系列数据变化采用场景的方式进行配置，合理地设计了对象点数据结构及其使用配置文件的组织方式，通过采用简便直观的配置工具极大地降低了仿真场景过程的配置工作。从场景的执行角度出发，场景执行使用多线程并行的方式也极大地优化了事故仿真的流程和事故场景的并发性，可以提高仿真效率。

附图说明

[0025] 图1、本发明实施例的主线程MainThread流程图。

[0026] 图2、本发明实施例的定时检测子线程OnTimerCheckThread流程图。

[0027] 图3、本发明实施例的场景执行子线程SceneExecThread流程图。

具体实施方式

[0028] 场景指的是特定条件、事件发生时导致的一系列顺序变化的数据集合。在城市轨道交通电力监控仿真培训系统中实现场景功能主要包括场景配置文件的结构设计、场景配置以及场景的后台实现方式，本专利着重介绍基于数据点对象的场景配置文件结构设计和基于多线程的后台场景实现方式，而对于场景配置模块，其主要功能是使用UI界面做配置，并按格式生成场景文件，因此不做太多介绍。

[0029] (一) 基于数据点对象的场景配置文件结构设计

[0030] 设计良好的数据组织方式可以为场景过程的配置、场景过程的后台执行提供更好的支持，而且数据组织作为场景功能的基本接口，其连接了场景的前台配置和后台执行两个过程，二者通过这种组织方式的配置文件进行交互，而且数据组织其逻辑的设计合理性也增强了场景功能的可扩展性。

[0031] 为了设计配置文件的结构，先给出如下的定义。

[0032] 定义1数据点对象：电力监控系统中数据一般以对象的形式存在，常见的数据点对象包括遥信点(DoublePoint对象)、遥测点(MeasuredValue对象)、遥控点(DoubleCommand对象)和遥脉点(IntegratedTotal对象)，每个点对象又包括了多个属性，每个属性对应一个值。表1以DoublePoint对象为示例给出了一个遥信点的属性和值。

[0033] 表1 DoublePoint对象示例

[0034]

属性名称	属性类型	属性意义	值
ObjectId	数值类型	遥信点 Id (由系统生成的唯一编号)	520651985
Name	字符串类型	遥信点名称	121 断路器
Description	字符串类型	遥信点描述	121
Address	数值类型	遥信点的 104 规约地址	449
FieldState	数值类型	遥信点的实时状态 (不带时标)	1 (分闸)
ManualState	数值类型	人工置入的遥信点状态	0 (故障)
HistoricalState	数值类型	遥信点的实时状态 (带时标)	1 (分闸)
State	数值类型	遥信点的实时状态 (跟随 FieldState、HistoricalState 两个属性中变化时间靠后的变化)	1 (分闸)

[0035] 定义2点变量Var:每个点变量Var包含一个alias (用于唯一标识该点变量)、id (数据点对象的ObjectId)、attribute (数值类型属性)、compareOperation (包括“>”,“<”,“==”,“!=”,“≥”,“≤”) 和一个value (数值类型,可以为已经定义过的其它点变量的alias,如果为alias,则表示取该alias点id的attribute值,即该alias点的compareOperation和value无意义)。每个点变量逻辑表达式最后运算的结果为一个Bool类型。

[0036] 点变量举例:

[0037] {"alias": "V1", "id": 518523517, "attribute": "FieldState", "compareOperation": "==", "value": "0"}, 即 {V1:518523517->FieldState==0}, 点变量V1表示数据点518523517的FieldState属性为0时结果为true, 否则为false;

[0038] {"alias": "V3", "id": 527972140, "attribute": "FieldValue", "compareOperation": ">", "value": "5"}, 即 {V3:527972140->FieldValue>5}, 点变量V3表示数据点527972140的FieldValue属性大于5时结果为true, 否则为false;

[0039] {"alias": "V4", "id": 527972141, "attribute": "FieldValue", "compareOperation": "!=", "value": "V3"}, 即 {V4:527972141->FieldValue!=527972140->FieldValue}, 点变量V4表示数据点527972141的FieldValue属性与527972140的FieldValue属性不相等时结果为true, 否则为false, 即对于V4忽略V3的compareOperation和value。

[0040] 定义3中间变量MidVar:每个中间变量包含一个alias (用于唯一标识该中间变量)、expression (不带括号的简单与或表达式,表达式的连接符只能是“&&”或“||”,表达式中只能出现已经定义过的点变量和中间变量以及连接符)、compareOperation (包括“==”,“!=”) 和一个value (Bool类型)。每个中间变量逻辑表达式最后运算的结果为一个Bool类型。

[0041] 中间变量举例:

[0042] {"alias": "M1", "expression": "V1&&V2", "compareOperation": "==", "value": "0"},

value":true,},即 {M1:V1&&V2==true},中间变量M1表示逻辑表达式“V1&&V2==true”的结果,即V1和V2的值都为true时M1的值为true,否则M1为false;

[0043] {"alias":"M2","expression":"M1&&V2","compareOperation":"==","value":true,},即 {M2:M1||V2==true},中间变量M2表示逻辑表达式“M1&&V2==true”的结果,即M1或者V2的为true时M2的值为true,否则M2为false;

[0044] 定义4场景步骤:每个场景步骤包含一个alias(用于唯一标识该场景步骤)、id(数据点对象的ObjectId)、attribute(数值类型属性)、value,以及一个event事件组合,event中可以包含有delay(用于表示在执行这个步骤前延时的毫秒时间)、condition(用于表示执行这个步骤的前提条件,一般配置为一个中间变量/点变量,当该中间变量/点变量的值为真时,表示这个步骤的前提条件得以满足,可以设置该点的值,若delay与condition同时存在,则先执行delay后再判断condition)、在配置了condition时还允许配置condition的检测周期conditionCycleCount与周期的时间conditionCyclePeriod毫秒(也可以不配置),表示在conditionCycleCount个周期内每隔conditionCyclePeriod时间检查condition,若condition为真则该步骤的条件满足可以设置该点的值,若一直检测不到该条件满足,则该步骤失败。

[0045] 场景步骤举例:

[0046] {"alias":"S2","id":518523516,"attribute":"FieldState","value":"1",
[0047] "event":{"condition":"M1","conditionCycleCount":20,"conditionCyclePeriod":1000,"delay":200,}},步骤S2表示在延时200ms后将在20个周期(每个周期1000ms)内循环检测中间变量M1的值,如果在周期内检测到其值为true则将518523516的FieldState属性设置为1;若一直检测不到值为true,则该步骤失败,场景执行失败。

[0048] 本发明中使用配置文件的形式来组织场景中出现的各种数据。每个配置文件中包含了场景的基本信息、场景步骤、以及场景步骤所使用的变量信息等等,基于定义1-4,设计的场景配置文件结构,文件使用JSON(JavaScript Object Notation)数据格式进行组织。

[0049] ①场景的基本信息:

[0050] ➤ scenceName:场景名称

[0051] ➤ scenceDescription:场景描述

[0052] ➤ sceneEnabled:是否启用场景,true启用,false停用

[0053] ➤ isCycledScene:是否为循环出,true循环场景,false非循环场景

[0054] ➤ onTimeExec:定时执行场景的时间,大于当前时间为有效,否则无效

[0055] ➤ onConditionExec:条件执行场景的条件表达式,可配置为中间变量/点变量,每次在条件表达式的结果从false变位true时自动执行一次场景。

[0056] ②点变量集合;

[0057] ③中间变量集合;

[0058] ④场景步骤集合。

[0059] (二)基于多线程的后台场景实现方式

[0060] 触发场景的执行方式一共有三种:

[0061] 1) 定时执行:要求场景配置为启用,即sceneEnabled=true,定时时间到后,立即执行场景,若场景为循环场景则一直在后台循环执行场景步骤;

[0062] 2) 条件执行:要求场景配置为启用,即sceneEnabled=true,后台循环检测场景的条件表达式,在每次条件表达式的结果从false变为true时执行一次场景。

[0063] 3) 立即执行:不要求场景配置为启用,前台的教师仿真HMI界面触发场景立即执行后直接将场景执行一次(即使该场景为循环场景也只执行一次)。

[0064] 所有场景的执行都受控于教学状态(开始、暂停、结束三种状态),只有教学状态为开始时才能执行,当教学状态设置为暂停时所有已经在执行中的场景会在执行完当前所执行的这一步骤之后暂停后面的执行步骤,等待教学状态再变为开始后继续执行,当教学状态设置为结束时结束所有场景的执行。

[0065] 基于多线程的后台场景实现方式中主要包括三种线程,主线程1个,定时检测子线程1个,场景执行子线程N个($0 \leq N$ 视同时执行的场景个数而定)。

[0066] 主线程MainThread是场景执行进程的后台主线程,其负责场景文件的解析与加载,监听所有场景文件是否需要立即执行,并负责启动定时检测子线程,其执行流程图如图1所示。

[0067] 定时检测子线程OnTimerCheckThread由主线程启动,其每隔1秒钟对定时类型的场景进行检测,定时时间到启动子线程执行该定时场景,同时对条件类型的场景条件表达式进行检测,若其条件表达式的值从false变为true则启动子线程执行该条件场景,需要注意的是对于循环类型的条件执行场景,并不会在条件为true时一直循环执行场景内容,而是每次条件从false变为true时执行一次,从而达到仿真实际故障发生时出现相应状态的效果。

[0068] 该线程的执行流程图如图2所示。

[0069] 场景执行子线程SceneExecThread负责场景内容的具体执行,其可由主线程、定时检测子线程启动,各个场景执行子线程之间相互不受影响。如图3所示,场景执行子线程SceneExecThread流程图。

[0070] 典型应用:

[0071] 作为本公司城市轨道交通电力监控培训仿真系统中的主要功能组件之一,本专利所发明的场景设计与实现方案已经在系统中得到了应用与验证。该培训仿真系统已成功用于南京、沈阳等多地的PSCADA培训仿真教学任务。

[0072] 以上的实施例仅为说明本发明的技术思想,不能以此限定本发明的保护范围,凡是按照本发明提出的技术思想,在技术方案基础上所做的任何非实质性改动,均落入本发明保护范围之内。

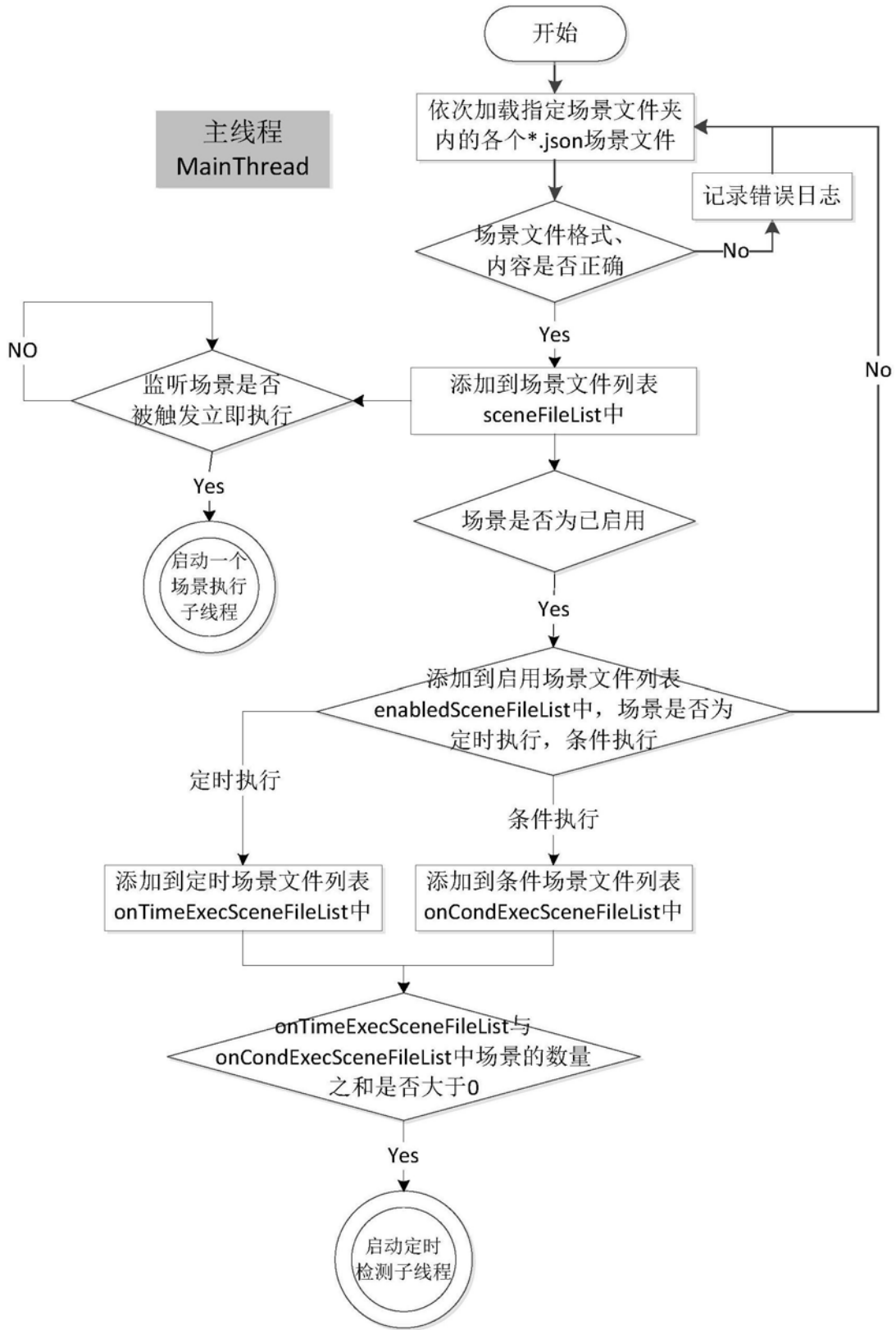


图1

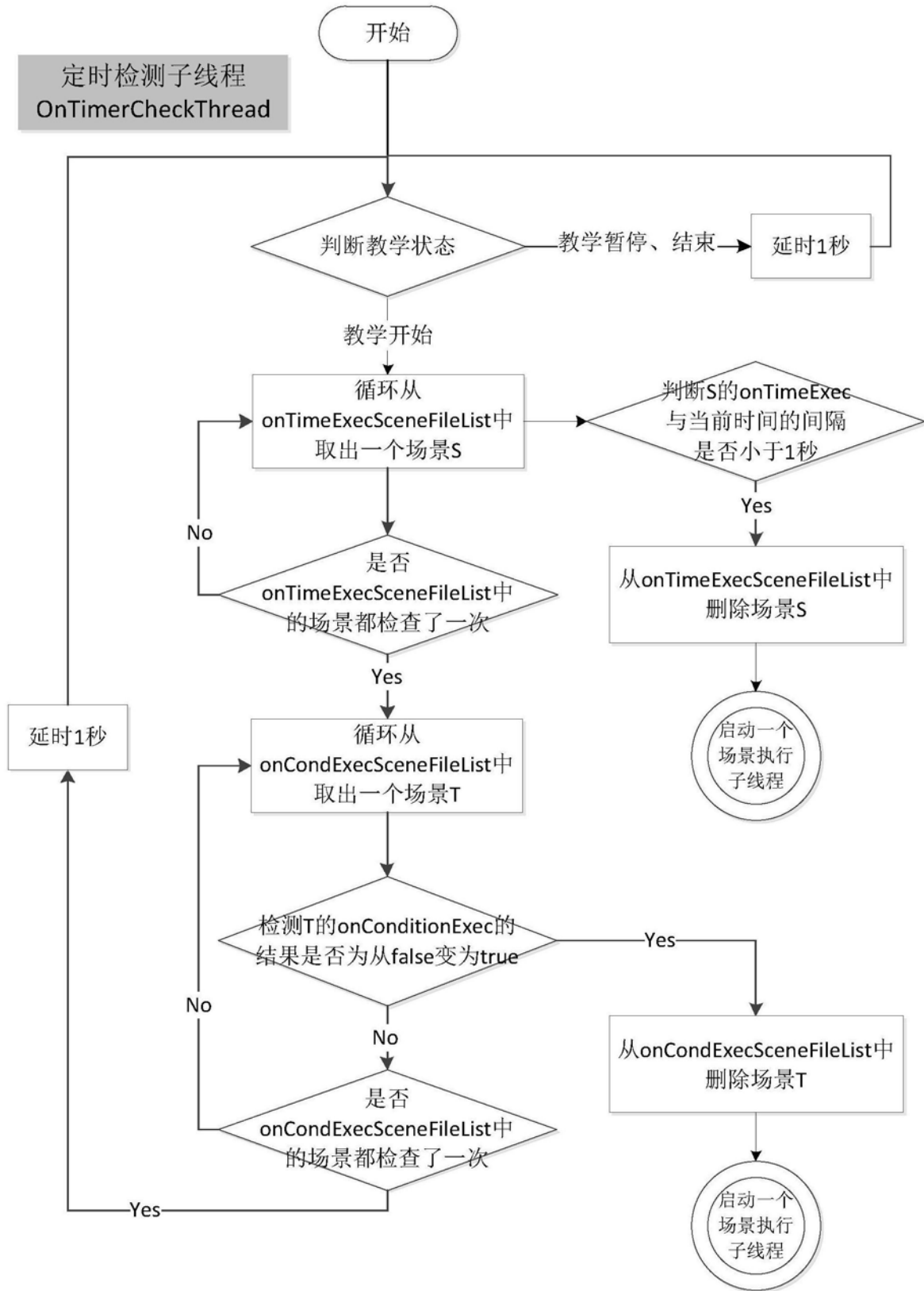


图2

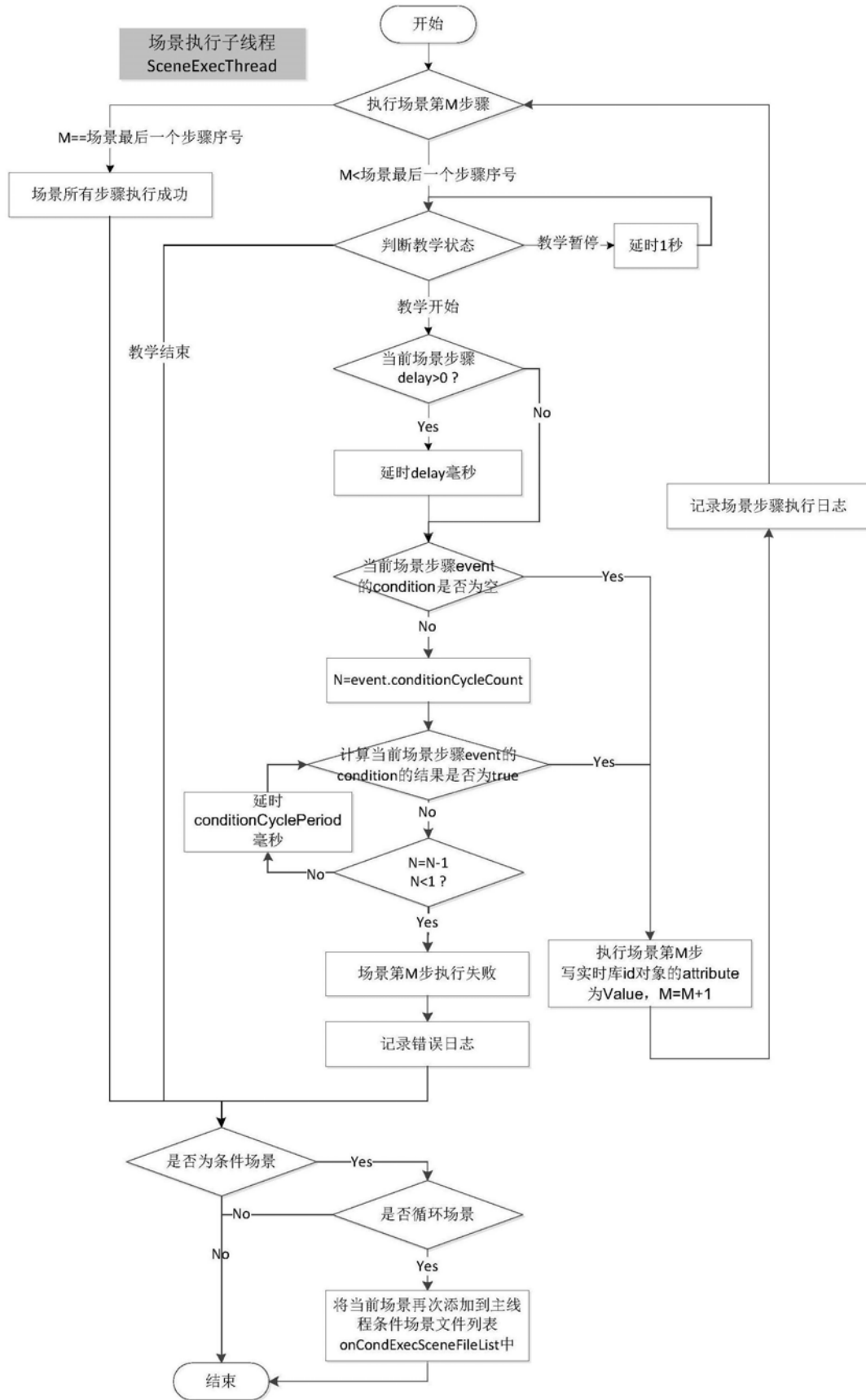


图3