武器评估子系统详细设计

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 修订时间 | 修订内容 | 修订人 |
| 0.1 | 2015-12-24 | 建立 | 曾辰熙 |
| 0.2 | 2015-12-30 | 初稿，除研发背景及附录未完成外，其他章节完成初稿 | 曾辰熙 |
| 0.2.1 | 2015-12-31 | 附录待补充 | 曾辰熙 |

目录

[1 研发背景 4](#_Toc439324081)

[1.1 需求 4](#_Toc439324082)

[1.2 编写目的 4](#_Toc439324083)

[2 总体设计 5](#_Toc439324084)

[2.1 概念定义 5](#_Toc439324085)

[2.2 设计思想 7](#_Toc439324086)

[2.3 总体架构 8](#_Toc439324087)

[2.4 功能边界 8](#_Toc439324088)

[2.4.1 指标配置 8](#_Toc439324089)

[2.4.2 节点配置 8](#_Toc439324090)

[2.4.3 边配置 9](#_Toc439324091)

[2.4.4 评估配置 9](#_Toc439324092)

[2.4.5 评估 9](#_Toc439324093)

[2.5 技术规格 9](#_Toc439324094)

[3 详细设计 11](#_Toc439324095)

[3.1 通用查询 11](#_Toc439324096)

[3.1.1 前端 11](#_Toc439324097)

[3.1.1.1 页面设计 11](#_Toc439324098)

[3.1.1.2 事件设计 11](#_Toc439324099)

[3.1.2 后端 12](#_Toc439324100)

[3.1.2.1 初始事件 12](#_Toc439324101)

[3.1.2.2 进入事件 14](#_Toc439324102)

[3.1.2.3 查找事件 15](#_Toc439324103)

[3.1.2.4 新增事件 17](#_Toc439324104)

[3.1.2.5 修改事件 18](#_Toc439324105)

[3.1.2.6 删除事件 18](#_Toc439324106)

[3.2 通用新增 20](#_Toc439324107)

[3.2.1 前端 20](#_Toc439324108)

[3.2.1.1 页面设计 20](#_Toc439324109)

[3.2.1.2 事件设计 20](#_Toc439324110)

[3.2.2 后端 21](#_Toc439324111)

[3.2.2.1 初始事件 21](#_Toc439324112)

[3.2.2.2 修改事件 21](#_Toc439324113)

[3.2.2.3 提交事件 22](#_Toc439324114)

[3.3 通用修改 23](#_Toc439324115)

[3.3.1 前端 23](#_Toc439324116)

[3.3.1.1 页面设计 23](#_Toc439324117)

[3.3.1.2 事件设计 24](#_Toc439324118)

[3.3.2 后端 24](#_Toc439324119)

[3.3.2.1 初始事件 24](#_Toc439324120)

[3.3.2.2 查找事件 25](#_Toc439324121)

[3.3.2.3 提交事件 25](#_Toc439324122)

[3.4 评估 26](#_Toc439324123)

[3.4.1 前端 26](#_Toc439324124)

[3.4.1.1 页面设计 26](#_Toc439324125)

[3.4.1.2 事件设计 27](#_Toc439324126)

[3.4.2 后端 27](#_Toc439324127)

[3.4.2.1 进入事件 27](#_Toc439324128)

[3.4.2.2 提交事件 28](#_Toc439324129)

[4 表结构设计 32](#_Toc439324130)

[4.1 指标配置表 32](#_Toc439324131)

[4.2 节点配置表 32](#_Toc439324132)

[4.3 边配置表 33](#_Toc439324133)

[4.4 评估配置表 33](#_Toc439324134)

[4.5 原始数据表 33](#_Toc439324135)

[4.6 指标数据缓存表 34](#_Toc439324136)

[4.7 试验信息表（测试用表） 34](#_Toc439324137)

[5 参考资料 35](#_Toc439324138)

[6 附录 36](#_Toc439324139)

[6.1 自定义校验函数 36](#_Toc439324140)

[6.2 自定义融合函数 36](#_Toc439324141)

[6.3 自定义近似函数 36](#_Toc439324142)

[6.4 常用无量纲函数 37](#_Toc439324143)

[6.5 常用隶属度函数 37](#_Toc439324144)

[6.6 常用综合函数 37](#_Toc439324145)

[6.7 配置文件 38](#_Toc439324146)

[6.7.1 指标配置文件 38](#_Toc439324147)

[6.7.2 节点配置文件 38](#_Toc439324148)

[6.7.3 边配置文件 38](#_Toc439324149)

[6.7.4 评估配置文件 38](#_Toc439324150)

# 研发背景

## 需求

武器评估子系统是运行控制评估分系统的三个子系统之一，其他两个子系统包括试验控制子系统，态势显示子系统。攻防武器评估子系统的功能为：基于采集的数据与知识库数据，对被测武器实行基本的效果评估，能够提供给用户灵活的自定义评估配置界面并且能够以图标形式显示评估过程。

## 1.2 编写目的

本文档以概要设计人员，开发人员为主要读者。概要设计人员需对该文档的需求覆盖性和准确性及设计合理性进行审核，开发人员需遵守该文档的相关开发规定和设计原则来展开开发工作。

# 2 总体设计

## 2.1 概念定义

|  |  |
| --- | --- |
| 概念 | 说明 |
| 武器评估 | 采用层次结构（树形）的评估模型对武器进行评估，叶节点为评估量，根节点为效果量。第k层节点到第k+1层节点的边的值为第k+1层节点对第k层节点的权值。当定义好各层权值后，从叶节点输入评估量，可由下至上地算出效果量。计算方式参见文献[1]，9.2.2小节。各层权值事先可由AHP（层次分析法）确定，见文献[1]，9.1.1小节。 |
| 评估量 | 指标经过无量纲处理后得到的值。 |
| 效果量 | 评估量经过层次评估过程后最终计算出来的值。 |
| 定量评估 | 对下层节点使用加权平均方式计算上层节点。 |
| 定性评估 | 对下层节点使用模糊集合技术计算上层节点。 |
| 节点类型 | 节点分为定量节点和定性节点，定性节点的值需要进行量化和归一化之后才能进行评估。 |
| 原始数据 | 指采集系统从网络中节点上采集并保存的原始数据。 |
| 融合过程 | 将多条原始数据合并成一条数据的过程。例：采集系统从网络中各节点采集“是否已感染病毒”的原始数据，该原始数据的条数等同于节点数，“1”表示感染，“0”表示未感染。融合后，将这些数据整合成新数据“感染病毒节点数”，该新数据定义为融合数据。 |



Figure 1 融合过程

|  |  |
| --- | --- |
| 概念 | 说明 |
| 近似过程 | 因为评估子系统与采集系统各自独立运行，故评估子系统不知道采集系统什么时间采集的数据，故有可能导致评估子系统指定时间的数据采集系统恰好没有（通常如此），而在此时间前后采集系统都有采集数据。故需要根据前后一段时间范围内的采集数据进行近似，得到评估子系统需要的数据。采集方法包括取均值，最值，中间值等。 |



Figure 2 近似过程

|  |  |
| --- | --- |
| 概念 | 说明 |
| 计算过程 | 某些简单的指标值就是近似数据，比如cpu利用率。但是仍有一部分指标需要将近似数据进行计算获得，例如比值类型（病毒感染率=感染病毒节点数/总节点数）或者差值类型（内存损耗=试验前内存利用率-试验后内存利用率）的数据。 |



Figure 3 计算过程

|  |  |
| --- | --- |
| 概念 | 说明 |
| 无量纲处理过程 | 在数据库中缓存的指标数据的主键为指标编号和试验编号，无量纲处理时，将从同一种指标编号下的指标数据中取值。 |

## 2.2 设计思想

本系统从功能抽象上来说可分为两个部分：配置和评估。指标，节点，边，评估配置均是对单一数据库表的增删改查操作。评估则是读取配置信息，以后序遍历的方式计算出评估得分。故可以采用一通用的增删改查模块来解决多个配置页面开发的问题。另，指标值计算采用用户自定义公式计算的方式实现，故需要设计并开发一些自定义的公式函数，见[附录](#_6_附录)。

## 2.3 总体架构



## 2.4 功能边界

### 2.4.1 指标配置

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 优先级 | 说明 | 备注 |
| 001 | 指标新增 | 高 |  |  |
| 002 | 指标修改 | 高 |  |  |
| 003 | 指标删除 | 高 |  |  |
| 004 | 指标查询 | 高 |  |  |

### 2.4.2 节点配置

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 优先级 | 说明 | 备注 |
| 101 | 节点新增 | 高 |  |  |
| 102 | 节点修改 | 高 |  |  |
| 103 | 节点删除 | 高 |  |  |
| 104 | 节点查询 | 高 |  |  |

### 2.4.3 边配置

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 优先级 | 说明 | 备注 |
| 101 | 边新增 | 高 |  |  |
| 102 | 边修改 | 高 |  |  |
| 103 | 边删除 | 高 |  |  |
| 104 | 边查询 | 高 |  |  |

### 2.4.4 评估配置

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 优先级 | 说明 | 备注 |
| 201 | 评估新增 | 高 |  |  |
| 202 | 评估修改 | 高 |  |  |
| 203 | 评估删除 | 高 |  |  |
| 204 | 评估查询 | 高 |  |  |

### 2.4.5 评估

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 优先级 | 说明 | 备注 |
| 301 | 评估 | 高 |  |  |

## 2.5 技术规格

该系统采用BS设计模式，涉及到前端设计，后端设计，数据库设计等等。以下列出可能使用的核心技术：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 所属 | 说明 | 备注 |
| 1 | html5 | 前端 |  |  |
| 2 | css | 前端 |  |  |
| 3 | javascript | 前端 |  |  |
| 4 | Ajax | 前端 |  |  |
| 4 | angularJS | 前端 | 重要的javascript库 |  |
| 5 | Echarts | 前端 | 图表库 |  |
| 6 | SpringMVC | 后端 | http服务框架 |  |
| 7 | Hibernate | 后端 | 数据库接口 |  |
| 8 | MySQL | 数据库 |  |  |
| 9 | IK-Expression | 后端 | 表达式解析JAVA库 |  |

# 3 详细设计

## 3.1 通用查询

### 3.1.1 前端

#### 3.1.1.1 页面设计

URL：[http://domain/weaponevaluate/admin.jsp?table=[table]](http://domain/weaponevaluate/admin.jsp?table=%5btable%5d)



#### 3.1.1.2 事件设计

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 事件 | 处理方式 | 说明 |
| 01 | 初始事件 | 异步后端 | 见下文 |
| 02 | 进入事件 | 前端/异步后端 | 见下文 |
| 02 | 查找事件 | 异步后端 | 见下文 |
| 03 | 上一页事件 | 异步后端 | 见下文 |
| 04 | 下一页事件 | 异步后端 | 见下文 |
| 05 | 跳转事件 | 异步后端 | 见下文 |
| 06 | 新增事件 | 同步后端 | 见下文 |
| 07 | 修改事件（族） | 同步后端 | 见下文 |
| 08 | 删除事件（族） | 异步后端 | 见下文 |

说明：

异步后端：指使用AJAX等异步JS技术，动态刷新部分网页的技术。

同步后端：指使用传统表单提交的方式，跳转到新网页，由新网页显示结果。

前端：不与后端进行交互，直接由前端JS代码进行处理。

### 3.1.2 后端

#### 3.1.2.1 初始事件

|  |  |
| --- | --- |
| 事件说明 | 打开查询页面后，首先需要进行初始事件处理，初始事件将返回“标题”内容，”查询层“内容，“显示层”内容。 |
| URL | [http://domain/weaponevaluate/init.jsp?table=[table]](http://domain/weaponevaluate/init.jsp?table=%5btable%5d) |
| 输出 | JSON格式数据。 |
| 异常 |  |

##### 3.1.2.1.1 输出数据说明

{

result:[00], //响应结果，00为成功

message:[OK], //响应信息

title:{ //此处定义了查询/新增/更新页面标题显示内容

list:[title], //查询

insert:[title], //新增

update:[title] //更新

},

count:10, //每页条数

columns:[ //此处定义了需要显示的表的各列信息

{

id:[col1], //列在数据库中定义的名称

name:[列1], //列在页面上显示的中文名

isPrimaryKey:[true],//是否为主键，在新增页面时，主键的列的控件触发onchange事件后，需要进行唯一性校验。

isNull:[true], //是否为空，在新增或修改页面时，主键的列的控件触发onchange事件后，需要进行是否为空校验。

type:[int], //列的数据类型，integer/double/boolean/string/date

length:[len], //列的长度

control:[text], //列在页面上显示的控件类型，text/date/combo box

check: //在新增或修改页面时，主键的列的控件触发onchange事件后所调用的自定义校验函数，见附录[自定义校验函数](#_6.1_自定义校验函数)。

[

{

function:[function1],

argument:[col1, col2…]

} //调用函数时，第一个参数为本控件的值，之后的参数为argument中列对应的控件的值。

…

]

isSelect:[true], //是否作为查询条件

selectType:[exact], //查询类型，exact/blur/range，分别表示精确查找，模糊查找，范围查找，如果查询类型为exact或range的话，对应的控件为control所指定的控件类型，如果查询类型为blur的话，对应的控件为text控件类型。

range: //控件类型为combo box的控件的候选内容。

{

type:[0/1], //0-固定值，1-动态值，为固定值时从static中取候选内容，为动态值时从dynamic中取key，value，table和query组sql语句从数据库中查询出候选内容。

static:

[

{key1:[value1]},

{key2:[value2]}

…

],

dynamic:

{

key:[col],

value:[col],

table:[table]

where:[col1, col2…]

}

},

default:[dft] //默认值，查询层控件或新增页面对应控件的默认值。

},

…

]

}

##### 3.1.2.1.2 流程



#### 3.1.2.2 进入事件

|  |  |
| --- | --- |
| 事件说明 | 若对应列的控件类型为combo box，那么需要获取其候选内容。处理方法见初始事件的响应结果。如果是动态获取的话，则使用以下url与后端通信。其中参数key填写key值，value填写value值，table填写table值，query中填写对应列的控件的值，格式为”col1:value1,col2:value2…”，列与列之间用逗号分隔，列名与值之间以冒号分隔，然后异步提交后端。查询完成后，返回查询结果。 |
| URL | [http://domain/weaponevaluate/range.jsp?key=[key]&value=[value]&table=[table]&query=[col1:value1,col2:value2...]](http://domain/weaponevaluate/range.jsp?key=%5bkey%5d&value=%5bvalue%5d&table=%5btable%5d&query=%5bcol1:value1,col2:value2...%5d) |
| 输出 | JSON格式数据。 |
| 异常 |  |

##### 3.1.2.2.1 输出数据说明

{

result:[00], //响应结果，00为成功

message:[OK], //响应信息

data:[ //此处定义了需要显示的表的各列信息

{

key:[key1], //在下拉控件中显示的key值

value:[value1] //在下拉控件中显示的value值

},

…

]

}

##### 3.1.2.2.2 流程



#### 3.1.2.3 查找事件

|  |  |
| --- | --- |
| 事件说明 | 初始事件完成后将第一次触发查找事件，点击查询按钮也将触发查找事件。将url中data参数赋值为对应显示层中显示的列，格式为”col1,col2…”，列与列之间用逗号分隔；query参数赋值为对应的查询层中的项，格式为”col1:value1,col2:value2…”，列与列之间用逗号分隔，列名与值之间以冒号分隔，然后异步提交后端。查询完成后，返回查询结果。上一页事件，下一页事件，跳转事件类似。 |
| URL | [http://domain/weaponevaluate/list.jsp?table=[table]&data=[col1,col2]&query=[col1:value1,col2:value2...]&begin=[x]&end=[y]](http://domain/weaponevaluate/list.jsp?table=%5btable%5d&data=%5bcol1,col2%5d&query=%5bcol1:value1,col2:value2...%5d&begin=%5bx%5d&end=%5by%5d) |
| 输出 | JSON格式数据。 |
| 异常 |  |

##### 3.1.2.3.1 输出数据说明

{

result:[00], //响应结果，00为成功，01为查询无结果

message:[OK], //响应信息

data:[ //此处定义了需要显示的表的各列信息

{id:[col1], //列在数据库中定义的名称

value:[value1] //值

},

…

]

}

##### 3.1.2.3.2 流程



#### 3.1.2.4 新增事件

|  |  |
| --- | --- |
| 事件说明 | 点击新增按钮后，将跳转至对应的新增页面 |
| URL | [http://domain/weaponevaluate/insert?table=[table]](http://domain/weaponevaluate/insert?table=%5btable%5d) |
| 输出 | 新增页面。 |
| 异常 |  |

##### 3.1.2.4.1 输出数据说明

见[通用新增](#_3.2_通用新增)。

##### 3.1.2.4.2 流程

见[通用新增](#_3.2_通用新增)。

#### 3.1.2.5 修改事件

|  |  |
| --- | --- |
| 事件说明 | 点击修改按钮后，将url中query参数赋值为对应的primary key的列，格式为”col1:value1,col2:value2…”，列与列之间用逗号分隔，列名与值之间以冒号分隔，然后跳转至对应的修改页面。 |
| URL | [http://domain/weaponevaluate/update?table=[table]&query=[col1:value1,col2:value2…]](http://domain/weaponevaluate/update?table=%5btable%5d&query=%5bcol1:value1,col2:value2…%5d) |
| 输出 | 修改页面。 |
| 异常 |  |

##### 3.1.2.5.1 输出数据说明

见[通用修改](#_3.3_通用修改)。

##### 3.1.2.5.2 流程

见[通用修改](#_3.3_通用修改)。

#### 3.1.2.6 删除事件

|  |  |
| --- | --- |
| 事件说明 | 点击删除按钮后，将url中query参数赋值为对应的primary key的列，格式为”col1:value1,col2:value2…”，列与列之间用逗号分隔，列名与值之间以冒号分隔，然后异步提交至后端。删除完成后，重新触发查找事件。 |
| URL | [http://domain/weaponevaluate/delete?=[table]&query=[col1:value1,col2:value2…]](http://domain/weaponevaluate/update?table=%5btable%5d&query=%5bcol1:value1,col2:value2…%5d) |
| 输出 | JSON格式数据。 |
| 异常 |  |

##### 3.1.2.6.1 输出数据说明

{

result:[00], //响应结果，00为成功

message:[OK] //响应信息

}

##### 3.1.2.6.2 流程



## 3.2 通用新增

### 3.2.1 前端

#### 3.2.1.1 页面设计

URL：[http://domain/weaponevaluate/insert?table=[table]](http://domain/weaponevaluate/insert?table=%5btable%5d)



#### 3.2.1.2 事件设计

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 事件 | 处理方式 | 说明 |
| 01 | 初始事件 | 异步后端 | 见下文 |
| 02 | 进入事件 | 前端/异步后端 | 见[进入事件](#_3.1.2.2_进入事件)。 |
| 03 | 修改事件 | 前端+异步后端 | 修改事件触发后，首先进行是否为空的校验，然后进行类型校验，再进行自定义函数校验（如果有定义），最后进行唯一性校验（如果是主键）。前三者为前端处理，唯一性校验为异步后端处理。 |
| 04 | 提交事件 | 异步后端 | 见下文 |

说明：

异步后端：指使用AJAX等异步JS技术，动态刷新部分网页的技术。

同步后端：指使用传统表单提交的方式，跳转到新网页，由新网页显示结果。

前端：不与后端进行交互，直接由前端JS代码进行处理。

### 3.2.2 后端

#### 3.2.2.1 初始事件

|  |  |
| --- | --- |
| 事件说明 | 打开新增页面后，首先需要进行初始事件处理，初始事件将返回“标题”内容，”数据层“内容。 |
| URL | [http://domain/weaponevaluate/init?table=[table]](http://domain/weaponevaluate/init?table=%5btable%5d) |
| 输出 | JSON格式数据。 |
| 异常 |  |

##### 3.2.2.1.1 输出数据说明

见[初始事件](#_3.1.2.1_初始事件)。

##### 3.2.2.1.2 流程

见[初始事件](#_3.1.2.1_初始事件)。

#### 3.2.2.2 修改事件

|  |  |
| --- | --- |
| 事件说明 | 修改事件的处理中最后一步需要进行唯一性校验。将url中query参数赋值为对应的primary key的列，格式为”col1:value1,col2:value2…”，列与列之间用逗号分隔，列名与值之间以冒号分隔。提交至后端，后端返回校验结果。 |
| URL | [http://domain/weaponevaluate/isuique?table=[table]&query=[col1:value1,col2:value2…]](http://domain/weaponevaluate/update?table=%5btable%5d&query=%5bcol1:value1,col2:value2…%5d) |
| 输出 | JSON格式数据。 |
| 异常 |  |

##### 3.2.2.2.1 输出数据说明

见[删除事件](#_3.1.2.6_删除事件)。

##### 3.2.2.2.2 流程



#### 3.2.2.3 提交事件

|  |  |
| --- | --- |
| 事件说明 | 提交事件触发后，将url中data参数赋值为对应的所有列，格式为”col1:value1,col2:value2…”，列与列之间用逗号分隔，列名与值之间以冒号分隔。提交至后端，后端返回提交结果。 |
| URL | [http://domain/weaponevaluate/insert/commit?table=[table]&data=[col1:value1,col2:value2…]](http://domain/weaponevaluate/update?table=%5btable%5d&query=%5bcol1:value1,col2:value2…%5d) |
| 输出 | JSON格式数据。 |
| 异常 |  |

##### 3.2.2.3.1 输出数据说明

见[删除事件](#_3.1.2.6_删除事件)。

修改成功后，弹出窗口让用户选择是否继续新增，还是返回查询页面。

##### 3.2.2.3.2 流程



## 3.3 通用修改

### 3.3.1 前端

#### 3.3.1.1 页面设计

URL：[http://domain/weaponevaluate/update?table=[table]&query=[col1:value1,col2:value2…]](http://domain/weaponevaluate/update?table=%5btable%5d&query=%5bcol1:value1,col2:value2…%5d)



#### 3.3.1.2 事件设计

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 事件 | 处理方式 | 说明 |
| 01 | 初始事件 | 异步后端 | 见下文 |
| 02 | 查找事件 | 异步后端 | 见下文 |
| 03 | 进入事件 | 前端/异步后端 | 见[进入事件](#_3.1.2.2_进入事件)。 |
| 04 | 修改事件 | 前端 | 修改事件触发后，首先进行是否为空的校验，然后进行类型校验，再进行自定义函数校验（如果有定义），最后进行唯一性校验（如果是主键）。三者皆为前端处理。主键列不允许修改。 |
| 05 | 提交事件 | 异步后端 | 见下文 |

说明：

异步后端：指使用AJAX等异步JS技术，动态刷新部分网页的技术。

同步后端：指使用传统表单提交的方式，跳转到新网页，由新网页显示结果。

前端：不与后端进行交互，直接由前端JS代码进行处理。

### 3.3.2 后端

#### 3.3.2.1 初始事件

|  |  |
| --- | --- |
| 事件说明 | 打开新增页面后，首先需要进行初始事件处理，初始事件将返回“标题”内容，”数据层“内容。 |
| URL | [http://domain/weaponevaluate/init?table=[table]](http://domain/weaponevaluate/init?table=%5btable%5d) |
| 输出 | JSON格式数据。 |
| 异常 |  |

##### 3.3.2.1.1 输出数据说明

见[初始事件](#_3.1.2.1_初始事件)。

##### 3.3.2.1.2 流程

见[初始事件](#_3.1.2.1_初始事件)。

#### 3.3.2.2 查找事件

|  |  |
| --- | --- |
| 事件说明 | 初始事件完成后将触发查找事件。将url中query参数赋值为primary key的列，格式为”col1:value1,col2:value2…”，列与列之间用逗号分隔，列名与值之间以冒号分隔，然后异步提交后端。查询完成后，返回查询结果。 |
| URL | [http://domain/weaponevaluate/list.jsp?table=[table]&query=[col1:value1,col2:value2...]&begin=1&end=1](http://domain/weaponevaluate/list.jsp?table=%5btable%5d&query=%5bcol1:value1,col2:value2...%5d&begin=1&end=1) |
| 输出 | JSON格式数据。 |
| 异常 |  |

##### 3.3.2.2.1 输出数据说明

见[查找事件](#_3.1.2.3_查找事件)。

##### 3.3.2.2.2 流程

见[查找事件](#_3.1.2.3_查找事件)。

#### 3.3.2.3 提交事件

|  |  |
| --- | --- |
| 事件说明 | 提交事件触发后，将url中query参数赋值primary key对应的列，data参数赋值为对应的有修改的列，格式为”col1:value1,col2:value2…”，列与列之间用逗号分隔，列名与值之间以冒号分隔。提交至后端，后端返回提交结果。 |
| URL | [http://domain/weaponevaluate/update/commit?table=[table]&query=[col1:value1,col2:value2…]](http://domain/weaponevaluate/update?table=%5btable%5d&query=%5bcol1:value1,col2:value2…%5d)&data=[col1:value1,col2:value2…] |
| 输出 | JSON格式数据。 |
| 异常 |  |

##### 3.3.2.3.1 输出数据说明

见[删除事件](#_3.1.2.6_删除事件)。

修改成功后，弹出窗口让用户选择是否继续修改，还是返回查询页面。

##### 3.3.2.3.2 流程



## 3.4 评估

### 3.4.1 前端

#### 3.4.1.1 页面设计

URL：<http://domain/weaponevaluate/evaluate.jsp>



#### 3.4.1.2 事件设计

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 事件 | 处理方式 | 说明 |
| 01 | 进入事件 | 前端/异步后端 | 见下文 |
| 02 | 提交事件 | 异步后端 | 见下文 |
| 03 | 单选事件 | 前端 | 单选中树的某一节点的“是否显示”选项时，该节点将显示在图显示层中。 |
| 04 | 修改事件 | 前端 | 选择不同的图表类型，图显示层显示不同的图表。 |

说明：

异步后端：指使用AJAX等异步JS技术，动态刷新部分网页的技术。

同步后端：指使用传统表单提交的方式，跳转到新网页，由新网页显示结果。

前端：不与后端进行交互，直接由前端JS代码进行处理。

### 3.4.2 后端

#### 3.4.2.1 进入事件

|  |  |
| --- | --- |
| 事件说明 | 使用以下url与后端通信。其中参数key填写值“id”，value填写值“name”，table填写值“evaluate”，query中填写为空，然后异步提交后端。查询完成后，返回查询结果。 |
| URL | <http://domain/weaponevaluate/range.jsp?key=id&value=name&table=evaluate&query=> |
| 输出 | JSON格式数据。 |
| 异常 |  |

##### 3.4.2.1.1 输出数据说明

见[进入事件](#_3.1.2.2_进入事件)。

##### 3.4.2.1.2 流程

见[进入事件](#_3.1.2.2_进入事件)。

#### 3.4.2.2 提交事件

|  |  |
| --- | --- |
| 事件说明 | 提交事件触发后，将url中id参数，iscleancache参数（0表示不清理缓存，1表示清理缓存）赋值为对应的控件，提交至后端，后端返回提交结果。前端将根据输出数据内容显示试验信息，树形图及相关图表。 |
| URL | [http://domain/weaponevaluate/evaluate/commit?id=[id]](http://domain/weaponevaluate/evaluate/commit?id=%5bid%5d)&iscleancache=[0/1] |
| 输出 | JSON格式数据。 |
| 异常 |  |

##### 3.3.2.3.1 输出数据说明

{

result:[00], //响应结果，00为成功

message:[OK], //响应信息

experimentinfo: //试验信息

{

id:[id], //试验编号

name:[name], //试验名称

weapontype:[0-Ddos], //武器类型：0-Ddos/1-蠕虫/2-木马

tester:[name], //试验名称

begintime:[time], //开始时间

endtime:[time] //结束时间

}

vertex:

[

{

id:[id], //节点编号

name:[name], //节点名称

type:[0-定量节点], //节点类型：0-定量节点/1-定性节点

sscore:[score], //字符型节点得分，定量节点表示为小数，定性节点表示为分数，例如：“x/y”，其意义为总共有y个等级，该节点为第x个等级。

fscore:[score] //数值型节点得分，定性节点为得分进行量化和无量纲之后的得分。

},

…

],

edge:

[

{

child:[id], //子节点编号

parent:[id], //父节点编号

weight:[weight] //权重

},

…

]

}

##### 3.3.2.3.2 流程

① 评估流程



② 原始数据提取流程

原始数据流程作为指标计算表达式中自定义函数实现，其函数定义为：

public List select(String id, String environmentid, Date begintime, Date endtime);

输入：

Id-原始数据id

environmentid-原始数据所在环境id

begintime-起始时间

endtime-结束时间

输出：

编号为id，所在环境为environmentid，在begintime至endtime范围内的原始数据，格式为：

[

[[data0, t0], [data1, t0]...], //t0时刻的原始数据集合

[[data0, t1], [data1, t1]...], //t1时刻的原始数据集合

…

]



③ 融合流程（融合方法不一样，对应流程也不一样）

融合流程作为指标计算表达式中自定义函数实现，见附录[自定义融合函数](#_6.2_自定义融合函数)，其函数定义为：

public List [function](List data);

输入：

data-原始数据

输出：

融合数据，格式为：

[

[data, t0], //t0时刻的融合数据

[data, t1] , //t1时刻的融合数据

…

]

④ 近似流程（近似方法不一样，对应流程也不一样）

近似流程作为指标计算表达式中自定义函数实现，见附录[自定义近似函数](#_6.3_自定义近似函数)，其函数定义为：

public Object [function](List data, date t);

输入：

data-融合数据

t-时间t

输出：

指标值

# 4 表结构设计

## 4.1 指标配置表

表名：index

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列 | 名称 | 类型 | 主键 | 可空 | 说明 |
| 01 | id | 指标编号 | char(8) | Y | N |  |
| 02 | name | 指标名称 | varchar(256) | N | N |  |
| 03 | expression | 表达式 | varchar(1024) | N | N | 指标计算表达式 |
| 04 | type | 数据类型 | char(1) | N | N | 0-string  1-double  2-boolean |

## 4.2 节点配置表

表名：vertex

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列 | 名称 | 类型 | 主键 | 可空 | 说明 |
| 01 | evaluateid | 评估编号 | char(8) | Y | N |  |
| 02 | id | 节点编号 | char(8) | Y | N |  |
| 03 | name | 节点名称 | varchar(256) | N | N |  |
| 04 | class | 类别 | char(1) | N | N | 0-效果量节点  1-准则节点  2-评估量节点 |
| 05 | type | 类型 | char(1) | N | N | 0-定量节点  1-定性节点 |
| 06 | indexid | 指标编号 | char(8) | N | Y | 评估量节点必填 |
| 07 | nmidx | 指标无量纲函数 | char(2) | N | Y | 评估量节点必填。  可选项见附录[常用无量纲函数](#_6.4_常用无量纲化函数)。  00-自定义。 |
| 08 | nmidxexp | 指标无量纲函数表达式 | varchar(1024) | N | Y | 当nmidx为00时必填 |
| 09 | level | 级别 | varchar(64) | N | Y | 定量节点的级别，用逗号分隔，例如 1,2,3,4表示4个级别，定量节点的得分将从这4个值从取得，级别为正整数，数字越大代表级别越高。定性节点必填。 |
| 10 | nmlvl | 级别无量纲函数 | char(2) | N | Y | 定性节点必填。  可选项见附录[常用无量纲函数](#_6.4_常用无量纲化函数)。  00-自定义。 |
| 11 | nmlvlexp | 级别无量纲函数表达式 | varchar(1024) | N | Y | 当nmlvl为00时必填 |
| 12 | mbsp | 隶属度函数 | varchar(1024) | N | Y | 表达式，定性节点必填。  见附录[常用隶属度函数](#_6.5_常用隶属度函数)。 |
| 13 | conclude | 综合函数 | char(2) | N | Y | 定性节点必填。  可选项见附录[常用综合函数](#_6.6_常用综合函数) |

## 4.3 边配置表

表名：edge

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列 | 名称 | 类型 | 主键 | 可空 | 说明 |
| 01 | evaluateid | 评估编号 | char(8) | Y | N |  |
| 02 | child | 子节点编号 | char(8) | Y | N |  |
| 03 | parent | 父节点编号 | char(8) | Y | N |  |
| 04 | weight | 边权重 | double(8,8) | N | N | 所有非负权值的和在区间[0,1]，当权值为负时，真实权值为用1减去非负权值的和求平均。 |

## 4.4 评估配置表

表名：evaluate

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列 | 名称 | 类型 | 主键 | 可空 | 说明 |
| 01 | id | 评估编号 | char(8) | Y | N |  |
| 02 | name | 评估名称 | varchar(256) | N | N |  |
| 03 | expid | 试验编号 | char(8) | N | N |  |

## 4.5 原始数据表

表名：raw

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列 | 名称 | 类型 | 主键 | 可空 | 说明 |
| 01 | id | 原始数据编号 | char(8) | Y | N |  |
| 02 | name | 原始数据名称 | char(8) | N | N |  |
| 03 | environmentid | 环境编号 | char(8) | Y | N |  |
| 04 | nodeid | 节点编号 | char(8) | Y | N |  |
| 05 | type | 数据类型 | char(1) | N | N | 0-string  1-double  2-boolean |
| 06 | value | 数据值 | varchar(256) | N | Y |  |
| 07 | time | 时间 | date | N | N |  |

## 4.6 指标数据缓存表

表名：cache

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列 | 名称 | 类型 | 主键 | 可空 | 说明 |
| 01 | evaluateid | 评估编号 | char(8) | Y | N |  |
| 02 | indexid | 指标编号 | char(8) | Y | N |  |
| 03 | type | 数据类型 | char(1) | N | N | 0-string  1-double  2-boolean |
| 04 | value | 数据值 | varchar(256) | N | Y |  |

## 4.7 试验信息表（测试用表）

表名：experiment

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列 | 名称 | 类型 | 主键 | 可空 | 说明 |
| 01 | id | 试验编号 | char(8) | Y | N |  |
| 02 | name | 试验名称 | varchar(256) | N | N |  |
| 03 | wptype | 武器类型 | char(1) | N | N | 0-Ddos  1-蠕虫  2-木马 |
| 04 | tester | 试验者 | varchar(256) | N | N |  |
| 05 | bgtime | 开始时间 | date | N | N |  |
| 06 | edtime | 结束时间 | date | N | N |  |

# 5 参考资料

[1] 网络攻击效果评估导论

# 6 附录

## 6.1 自定义校验函数

|  |  |
| --- | --- |
| 01 | func1 |
| 说明 |  |
| 输入 |  |
|  |  |
|  |
| 输出 |  |
|  |  |
|  |
| 异常 |  |
|  |  |
|  |
| 02 | func2 |
| 说明 |  |
| 输入 |  |
|  |  |
|  |
| 输出 |  |
|  |  |
|  |
| 异常 |  |
|  |  |
|  |

## 6.2 自定义融合函数

|  |  |  |
| --- | --- | --- |
| 序号 | 名称 | 说明 |
| 01 | func1 |  |
|  |  |  |

## 6.3 自定义近似函数

|  |  |  |
| --- | --- | --- |
| 序号 | 名称 | 说明 |
| 01 | func1 |  |
|  |  |  |

## 6.4 常用无量纲函数

|  |  |  |
| --- | --- | --- |
| 序号 | 名称 | 说明 |
| 01 | 函数1 |  |
|  |  |  |

## 6.5 常用隶属度函数

|  |  |  |
| --- | --- | --- |
| 序号 | 名称 | 说明 |
| 01 | 函数1 |  |
|  |  |  |

## 6.6 常用综合函数

综合函数定义为：

public int [function](List vetor, List level);

输入：

vector-隶属度向量

level –级别数组

输出：

级别

|  |  |  |
| --- | --- | --- |
| 序号 | 名称 | 说明 |
| 01 | func1 |  |
|  |  |  |

## 6.7 配置文件

### 6.7.1 指标配置文件

### 6.7.2 节点配置文件

### 6.7.3 边配置文件

### 6.7.4 评估配置文件