武器评估子系统概要设计

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 修订时间 | 修订内容 | 修订人 |
| 0.1 | 2015-12-24 | 建立 | 曾辰熙 |

目录

[1 研发背景 4](#_Toc439077737)

[1.1 需求 4](#_Toc439077738)

[1.2 编写目的 4](#_Toc439077739)

[2 总体设计 5](#_Toc439077740)

[2.1 概念定义 5](#_Toc439077741)

[2.2 设计思想 8](#_Toc439077742)

[2.3 总体架构 8](#_Toc439077743)

[2.4 功能边界 8](#_Toc439077744)

[2.4.1 指标配置 8](#_Toc439077745)

[2.4.2 节点配置 9](#_Toc439077746)

[2.4.3 边配置 9](#_Toc439077747)

[2.4.4 评估配置 9](#_Toc439077748)

[2.4.5 评估 9](#_Toc439077749)

[2.5 技术规格 9](#_Toc439077750)

[3 详细设计 10](#_Toc439077751)

[3.1 通用查询 10](#_Toc439077752)

[3.1.1 前端 10](#_Toc439077753)

[3.1.1.1 页面设计 10](#_Toc439077754)

[3.1.1.2 事件设计 10](#_Toc439077755)

[3.1.2 后端 11](#_Toc439077756)

[3.1.2.1 初始事件 11](#_Toc439077757)

[3.1.2.2 进入事件 13](#_Toc439077758)

[3.1.2.3 查找事件 14](#_Toc439077759)

[3.1.2.4 新增事件 15](#_Toc439077760)

[3.1.2.5 修改事件 15](#_Toc439077761)

[3.1.2.6 删除事件 16](#_Toc439077762)

[3.2 通用新增 17](#_Toc439077763)

[3.2.1 前端 17](#_Toc439077764)

[3.2.1.1 页面设计 17](#_Toc439077765)

[3.2.1.2 事件设计 17](#_Toc439077766)

[3.2.2 后端 18](#_Toc439077767)

[3.2.2.1 初始事件 18](#_Toc439077768)

[3.2.2.2 修改事件 18](#_Toc439077769)

[3.2.2.3 提交事件 19](#_Toc439077770)

[3.3 通用修改 20](#_Toc439077771)

[3.3.1 前端 20](#_Toc439077772)

[3.3.1.1 页面设计 20](#_Toc439077773)

[3.3.1.2 事件设计 20](#_Toc439077774)

[3.3.2 后端 21](#_Toc439077775)

[3.3.2.1 初始事件 21](#_Toc439077776)

[3.3.2.2 查找事件 21](#_Toc439077777)

[3.3.2.3 提交事件 22](#_Toc439077778)

[3.4 评估 23](#_Toc439077779)

[3.4.1 前端 23](#_Toc439077780)

[3.4.1.1 页面设计 23](#_Toc439077781)

[3.4.1.2 事件设计 24](#_Toc439077782)

[3.4.2 后端 24](#_Toc439077783)

[3.4.2.1 进入事件 24](#_Toc439077784)

[3.4.2.2 提交事件 25](#_Toc439077785)

[4 参考资料 26](#_Toc439077786)

# 研发背景

## 1.1 需求

## 1.2 编写目的

# 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 | 指标经过无量纲化处理后得到的值。 |
| 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 |
| 无量纲化处理过程 | 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，无量纲化处理时，将从同一种指标编号下的指标数据中取最值。 |

## 2.2 设计思想

## 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 |
|  |  |  |  |  |

# 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。

[

{

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中列对应的空间的值。

…

]

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]&query=[col1:value1,col2:value2...]&begin=[x]&end=[y]](http://domain/weaponevaluate/list.jsp?table=%5btable%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

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

#### 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

#### 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

#### 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

#### 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

#### 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

#### 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

score:[score] //节点得分

},

…

],

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

# 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 |

## 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 | varchar(64) | N | Y | 评估量节点必填 |
| 08 | 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。 |
| 09 | nmlvl | 级别无量纲函数 | varchar(64) | N | Y | 定性节点必填。 |
| 10 | mbsp | 隶属度函数 | varchar(1024) | N | Y | 定性节点必填。 |

## 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 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 配置文件设计

## 5.1 指标配置文件

## 5.2 节点配置文件

## 5.3 边配置文件

## 5.4 评估配置文件

# 6 代码设计

# 5 参考资料

[1] 网络攻击效果评估导论