

# 信息系统应用关系分析器<sup>①</sup>

党执政<sup>1</sup>, 廉东本<sup>2</sup>

<sup>1</sup>(中国科学院大学 沈阳计算技术研究所, 沈阳 110168)

<sup>2</sup>(中国科学院 沈阳计算技术研究所, 沈阳 110168)

**摘要:** 信息系统应用关系分析器是一个以三维可视化的方式分析信息系统的各应用之间关系的工具。可以以三维可视化的方式直观的展示所有应用节点、应用之间的关联关系、流向关系、应用基本信息、应用监控信息和应用告警信息等, 并且能够在某一应用节点出现问题时直观、快速、准确的定位故障节点以及影响范围。同时分析器还可以以三维可视化的方式展示应用节点所包含的内部系统结构以及各结构之间的关联关系。分析器数据来源来自关系数据库, 后端使用 J2EE 技术, 三维可视化使用 Unity3D。整个分析器具有部署灵活、扩展性好、操作简便、数据呈现清晰直观等特性。

**关键词:** 信息系统; 应用关系; 智能布局; 智能分析; 可视化

## Application Relation Analyzer for Information System

DANG Zhi-Zheng<sup>1</sup>, LIAN Dong-Ben<sup>2</sup>

<sup>1</sup>(Shenyang Institute of Computing Technology, University of Chinese Academy of Sciences, Shenyang 110168, China)

<sup>2</sup>(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

**Abstract:** The application of the information system is a tool to analyze the relationship between the application of the information system in the way of 3D visualization. It can display all the application nodes, the relationship between the applications, the flow of information, the application of basic information, application of monitoring information and the application of alarm information. And it can be used to directly, quickly and accurately locate the fault nodes and the influence range when a problem occurs. At the same time, it also can display the relationship between the internal system structure and the structure of the application nodes in 3D visualization. Analyzer data source is from the relational database, back-end uses J2EE technology, three-dimensional visualization uses Unity3D. The whole analyzer has the characteristics of flexible deployment, good expansibility, easy operation, clear and intuitive data.

**Key words:** information system; application relationship; intelligent layout; intelligent analysis; visualization

## 1 引言

信息系统是由计算机硬件、网络设备、计算机软件、信息资源组成的系统<sup>[1]</sup>。

由于信息技术的迅速发展, 信息系统已经普及并渗入到社会的各个领域和各行各业之中。信息系统在电力、通信、电子商务、银行、数字城市等领域发挥着越来越重要的作用, 与人们的生活密不可分<sup>[2]</sup>。当前国内外大多数信息系统是以人工的方式来查看、分析、判断应用之间的关系, 不能快速、准确的定位故

障节点, 不能快速、准确的分析故障节点对其它相关节点的影响程度, 而且人工成本很高, 效率低下<sup>[3]</sup>。鉴于此, 本文设计了信息系统应用关系分析器。能够准确、自动地展示并智能的分析信息系统中的应用节点关联关系及状态信息。

信息系统应用关系分析器具有重要应用意义, 首先, 分析器以一种直观的可视化方式展示应用之间以及应用所属之间的关联关系, 方便用户对相应的信息系统进行梳理。其次, 当某个应用节点出现故障或者

① 基金项目: 国家水体污染控制与治理科技重大专项(2012ZX07505003)

收稿时间: 2015-11-05; 收到修改稿时间: 2015-12-20 [doi:10.15888/j.cnki.csa.005230]

应用节点进行变更时可以对故障或变更节点进行快速定位,并迅速分析故障或变更的影响程度和范围<sup>[4]</sup>.

## 2 分析器体系结构与功能介绍

### 2.1 分析器体系结构

分析器体系结构主要分为数据服务层、数据处理层和可视化分析层,如图1所示.

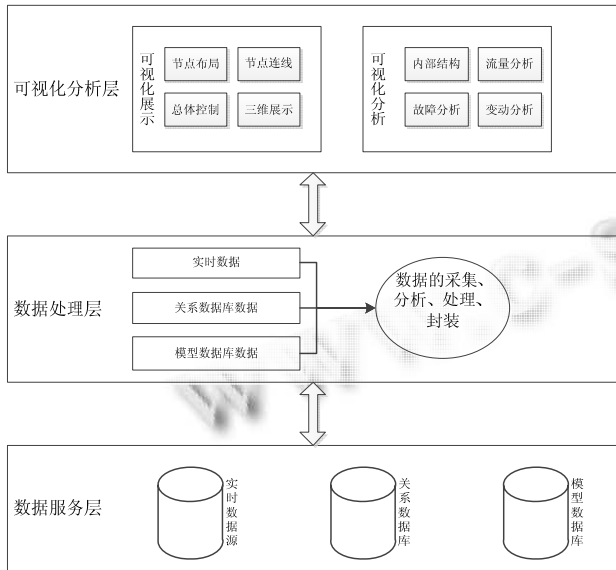


图1 分析器体系结构

#### (1)数据服务层

数据服务层是整个分析器的数据来源.数据服务层包括实时数据、关系数据库数据、模型数据.实时数据主要包括信息系统中应用的实时性监控信息、报警信息等.关系型数据库中主要存储应用的基本信息、应用之间的连接与所属信息、应用的拓扑布局信息等.模型数据库主要提供三维展示时的模型实例.

#### (2)数据处理层

数据处理层主要对数据服务层提供的各种数据进行采集、分析、处理和封装.关系数据库中的数据是主要的数据来源,因此需要对数据进行深加工,把数据库中的数据加工处理并存储在所需的数据结构之中.模型数据库中的数据根据需求直接调用.

#### (3)可视化分析层

可视化分析层主要是对信息系统中应用之间的层次以及所属关系以可视化的方式进行展示,为分析应用之间关系提供可视化依据.三维展示层用3D引擎Unity3D实现,主要包括可视化展示和可视化分析

两个组成部分.

### 2.2 分析器功能介绍

分析器主要功能模块包括控制模块、分析模块和数据模块,如图2所示.

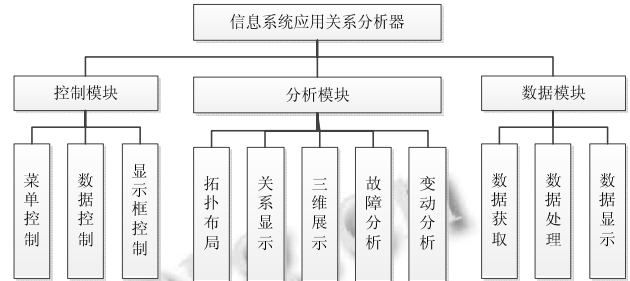


图2 分析器主要功能模块

#### (1)控制模块

控制模块对整个分析器起到整体控制的作用,主要有菜单控制、数据驱动控制和显示框控制三种方式.通过控制模块可以直观、方便的对分析器进行操作.

#### (2)分析模块

分析模块是整个分析器的核心模块,主要包括拓扑布局、关系显示、三维展示、故障分析和变动分析.通过分析模块,分析器可以智能的对应用节点进行布局,动态的显示应用之间的连接与所属关系,以三维的方式展示整个信息系统的的关系.当出现节点故障或者节点变动时可以快速的对故障以及变动节点进行定位,智能分析变动节点或者故障节点对整个信息系统造成的影响.

#### (3)数据模块

数据模块为整个分析器提供数据支持,分析器可以动态的获得所需数据,并且对获得的数据进行分析、处理和封装,以供分析器其它模块使用,对需要显示的数据可以直接在三维场景进行显示.

## 3 分析器主要技术说明

### 3.1 数据交互

整个分析器的Web端是用J2EE开发的,而三维展示是通过Unity3D展示的.Web端和Unity3D之间的数据交互显得尤为重要.Unity3D与Web通信主要是通过Unity3D与JavaScript的通信,而且通信的内容只能是以字符串的方式.因此,我们选择用JSON对数据进行封装,然后使用JSON字符串进行通信.JavaScript

与 Unity 通信主要有一下两个方面:

(1)JavaScript 调用 Unity 内容.

JavaScript 调用 Unity 的内容需要使用 SendMessage()函数, 该函数需要由 Unity 的相关对象调用, UnityEngine.Object 类提供了 SendMessage()函数提供支持, 以下是实现代码:

```

var u = new UnityEngine.Object2();
u.initPlugin(jQuery("#unityPlayer")[0], "a.unity3d");
u.getUnity().SendMessage(objectName, functionName, stringName);

```

(2)Unity 调用 JavaScript

通过 Unity 提供的系统函数 Application.ExternalCall("unityCall", "unityCallSuccess"); 该函数的第一个参数是 JavaScript 的函数名, 第二个参数是调用该函数传递的参数值, 即需要传输的 JSON 字符串.

### 3.2 拓扑布局算法

分析器的三维可视化展示部分主要分为两个层次, 如图 3 所示. 本节主要提出适合本分析器的多层次布局算法以及连线规则.

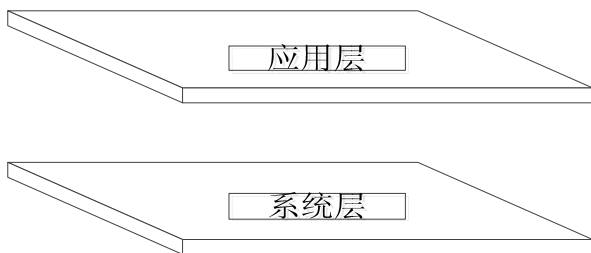


图 3 分析器三维可视化层次图

#### 3.2.1 分析器拓扑布局算法的选择

现有的布局算法主要包括树型布局算法、射线型布局算法、层次性布局算法、直交布局算法和混合型布局算法等<sup>[5-6]</sup>. 通过对现有布局算法的分析发现这些布局算法都有自己的使用范围, 选择其中任何一种都不能很好的完成设计的目标. 由于应用关系以及所属关系的复杂性, 整个三维可视化部分的拓扑布局采用混合型布局算法, 分别为应用层、系统层. 在每一层中分别采用不同的改进的拓扑布局算法.

#### 3.2.2 各层次布局算法描述

(1)应用层布局算法描述

应用层主要展示的是应用以及应用之间的关联与流向关系. 这一层我们使用改进的直交布局算法. 算法描述如图 4 所示<sup>[5-6]</sup>.

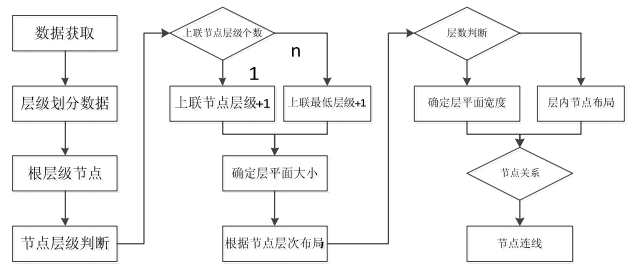


图 4 应用层布局算法描述图

首先从数据中获取所有应用节点信息, 把应用节点按层级划分, 根据应用之间的流向分为不同的层级, 从左到右依次为 1-n 级. 首先找到最开始的根层级, 从根节点算起, 找到与它相关的下一级节点, 放在第二个层级上. 以此类推, 把第二个层级的节点当做根节点来找出各节点的下一层级的所有节点放在当前节点的下一层级上. 这里有个节点所在层级判断特别注意, 如果一个节点的上联节点分别在多个层级上, 这时候我们选取这些上联节点所述层级的最低的一个层级, 然后当前节点在最后一个层级的下一层级. 节点所属层级划分完之后, 根据层数和每层的节点数确定层平面的大小, 然后将不同层以及同一层级的节点按照从左到右、从上到下均匀的排列. 以下为部分核心源代码:

```

//用于应用层布局的节点信息
private Dictionary<int, List<string>> appPointsInfo;
private float levelCount //保存层级数;
private float maxLevelCount; //保存每层节点个数
//节点布局
for (int i=1; i<=levelCount; i++) {
    appName=appPointsInfo[i];
    appX=disX/2f+(i-1)*10;
    disZ=maxLevelCount*10f/appName.Count;
    for(int j=0;j<appName.Count;j++){
        appZ=disZ/2f+j*disZ;
        GameObject a = (GameObject)Instantiate
(app,newVector3(0,0,0),app.transform.rotation);
        a.transform.position=new
Vector3(appX,cloneApp.transform.position.y,appZ);
    }
}

```

考虑完节点的布局, 我们讨论节点之间的连线关系. 因为节点都是按照层级划分的, 所以在连线时只

要根据相邻的两个层级的节点关系直接连接。如果一个节点跟多个节点之间有所属关系，那么我们考虑用三维的方式进行连线，以避免线条过于混乱。

(2) 系统层布局算法

系统层主要展示的是单个应用所包含的多个系统的所属关系。这一层我们选择变形的树型拓扑布局算法。算法描述如图5所示<sup>[7-9]</sup>。

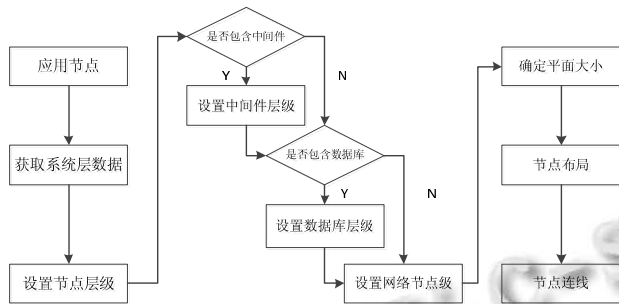


图5 系统层布局算法描述图

应用层的每个节点与系统层的节点是一对多的关系，一个应用可能包含多个系统结构，这些系统都有共同的父节点。所以可以把应用层的节点当做树的根节点。系统层的其他节点按照不同的层级作为叶子节点。系统层的层级分为系统层级、中间件层级、数据库层级和网络节点层级。首先把应用的所有系统放在系统层级，从左到右依次布局，然后判断系统是否包含中间件，如果包含，把所对应的中间件放到中间件层级，如果不包含不作处理。接下来判断不包含中间件的系统和中间件是否包含数据库，如果包含则放在数据库层级，如果不包含则不作处理，最后把各节点相关的网络节点放置在网络节点级。根据层级数和层内节点数确定平面大小，然后进行节点布局和节点间连线。以下为部分核心代码：

//用于系统层布局的节点信息

```
private Dictionary<string,List<string>> sysPointsInfo;
private float sysLevelCount; //保存层级数;
private float sysMaxLevelCount; //保存每层节点个数
for (int i=0; i<sysType.Length; i++) {
    sysName=sysPointsInfo[sysType[i]];
    appY=disY/2f+i*10;
    disX=sysMaxLevelCountSize*10f/sysName.Count;
    for(int j=0;j<sysName.Count;j++){
        appX=disX/2f+j*disX;
```

```
GameObject a = (GameObject)Instantiate
(sysPoint, sysPointStartPos, appPoint.transform.rotation);
a.transform.position=new
Vector3(sysPointStartPos.x+appX,sysPointStartPos.y-app
Y-10f,cloneSysLevel.transform.position.z);
}
}
```

连线规则使用传统的树型直连规则，对于不同层级连接同一层级节点的情况，我们可以在每一个层级上设置一个拐点层，所有需要连接本层级的连线都通过这个拐点层进行连接。这在一定程度上减少了连线的复杂性和混乱性。

3.3 分析器实现流程

分析器根据不同的数据，按照不同层次的不同布局算法进行三维可视化布局并展示，为应用之间的可视化分析提供依据。具体步骤如下：

- ① 数据封装：主要是对数据进行采集，包括从数据库数据和实时数据，把采集到的数据根据设计好的数据结构进行封装，数据结构中包含三维展示的所有信息。
- ② 数据交互：根据 3.1 节制定的交互路线，把 Web 端封装好的数据传输给 Unity3D，为可视化部分提供数据支持。
- ③ 节点布局：以 Web 端传输的数据为基础，根据分析器不同层次的拓扑布局算法，将应用节点按照布局算法布局到相应层次的位置上，使应用节点清晰、均衡的显示在不同层次中。
- ④ 节点连线：根据应用之间的关联关系和所属关系将不同和相同层次中的节点按照相应的连线规则进行节点之间的连线，通过连线的方式显示应用之间的层次以及流向关系。
- ⑤ 可视化分析：根据节点的布局算法和连线规则，使应用之间的关联关系和所属关系以三维可视化的方式展示，为整个信息系统的可视化显示、应用节点的基本信息显示以及应用节点发生故障或出现变更时的影响评估提供可视化依据。

4 分析器运行效果

信息系统应用关系分析器的运行效果如图6所示。



图6 分析器运行效果图

分析器以三维可视化的方式展示了各应用之间的关联关系以及所属关系,各层节点布局清晰、灵活,直观的反应了应用之间的层次关系以及应用的所属结构。运行效果图中的红色部分准确定位了故障节点在整个应用层的位置,为快速分析故障影响提供了可视化依据,并以弹出框的方式显示应用的相关基本信息。实际使用发现分析器具有很好的灵活性和可扩展性,操作简便直观,对于信息系统应用关系的分析具有很好的应用价值。

## 5 结语

本文论述了信息系统应用关系分析器的整体体系结构和分析器实现的主要技术。该分析器的特点是以三维可视化的方式直观的展示所有应用节点、应用之间的关联关系以及所属关系、应用基本信息、应用监控信息和应用告警信息等。当某个应用节点出现故障

时可以对故障节点进行快速定位,并迅速分析故障或变更的影响程度和范围。采用分析器可以极大的提高信息系统的可维护性和扩展性,同时降低了用户维护任务的人力和物力成本。

## 参考文献

- 1 王天新.管理信息系统发展.现代情报,2007,27(6):224-225.
- 2 陈佳乐.H3C 金融数据中心解决方案与实践.金融电子化,2009,(2):86-86.
- 3 祁进林.浅谈企业信息系统维护与管理.现代营销:学苑版,2011,7(7):374-374.
- 4 Shneiderman B. The eyes have it: A task by data type taxonomy for information visualizations. Proc. of IEEE Symposium on Visual Languages. IEEE. 1996. 336-343.
- 5 张伟明,罗军勇,王清贤.网络拓扑可视化研究综述.计算机应用研究,2008,25(6):1606-1610.
- 6 刘金明,万明祥.基于节点度分层的路由器级拓扑布局算法.计算机技术与发展,2015,25(1):100-106.
- 7 徐春燕,康建初,金毅,黄超.针对大型局域网拓扑图的布局算法.微计算机信息,2009,30(2):93-95.
- 8 Clauset A, Moore C. Why Mapping the Internet is Hard. eprint arXiv:cond-mat/0407339, 2004.
- 9 王聪,王翠荣,王兴伟,蒋定德.面向云计算的数据中心网络体系结构设计.计算机研究与发展,2012,49(2):286-293.