

语流中录制合成音节单元的策略,对系统的合成质量有较明显的改进。

### 3.4 抽取有效语音单元

以音节为合成单位进行波形连接的文语转换方法,音节单元录制的质量对系统的总体效果有重大的影响,本系统采用了以下策略以保证语音库音节单元的质量,即重复录音,概率择优<sup>[6]</sup>。对于每个音节单元,录制8遍,在建立合成音节库时,选取其中一个作为音节单元。选取的策略如下:

a) 舍弃时长小于所有该类语音单元平均时长80%的语音单元。这样处理的目的是,一方面,这些被删除的语音单元的音质往往不太好。另一方面,在

合成时如果使用这些被删除的语音单元,在调整时长时,时长的增加有可能超过限定的时长增长因子(目标语音单元时长/原始语音单元时长)的阈值。造成语音信号的失真或不自然。在本系统中,时长增长因子设为1.22。

b) 计算所有语音单元平均采样能量,舍弃平均采样能量大于该值的语音单元。这样处理可以减小在合成时声音的剧烈起伏振荡。

c) 对于进行以上两个步骤后剩下的语音单元,通过试听和合成测试,最后,在每一类型语音单元选定一个作为系统使用的语音单元。

(下转第89页)

(上接第4页)

### 主要参考文献

- [1] H. T. Kung et al., Systolic arrays (for VLSI), Sparse Matrix Proc. Society for Industrial and Applied Mathematics, 1978
- [2] H. T. Kung, Why systolic arrays, IEEE Computer, vol. 15 Jan. 1982
- [3] S. Hauck, Multi-FPGA Systems, PH. D Thesis, Dept. CS&E, University of Washington, 1995
- [4] B. K. Fawcett et al., Reconfigurable Processing with Field Programmable Gate Arrays, in Inter. Conf. Application Specific Array Processors, 1996
- [5] M. Annaratone, et al., The Warp Computer: Architecture, Implementation and Performance, IEEE Trans. Computer, 36(12)1987
- [6] 杨超峰, 颜玲, 傅宇卓, 胡铭曾, 支持 MPEG-2 标准的 ME 芯片设计, 体系结构年会'97, 1997. 10
- [7] W. Shang et al., On uniformation of affine dependence algorithms, IEEE Trans. Computer, 45 (7) 1996
- [8] Y. Wong et al., Transformation of broadcast into propagation in systolic arrays, J. of Parallel and Distributed Computing, 14(2)1992
- [9] J.-C. Tsay et al., Design of efficient regular arrays for matrix multiplication by two step regularization, IEEE Trans. Parallel Distrib. System, 6(2)1995
- [10] W. Li et al., A singular loop transformation framework based on non-singular matrices, Int. J. Parallel Programing, 22(2)1994
- [11] H. J. Lee et al., Automatic generation of modular mapping, ASAP'96, 1996
- [12] D. I. Moldovan et al., Partitioning and mapping algorithms in fixed size systolic arrays, IEEE Trans. Comput, vol. C-35, 1986
- [13] W. Shang et al., On time mapping of uniform dependence algorithms into lower dimensional processor arrays, IEEE Trans. Parallel and Distributed Systems, 3(3)1992
- [14] S. Y. Kung, VLSI Array Processors, Englewood Cliffs, NJ: Prentice Hall, 1988
- [15] G. -J. Li et al., The design of optimal systolic arrays, IEEE Trans. Comput. vol. C-34, 1985
- [16] K. Ganapathy et al., Optimal Synthesis of Algorithm-specific Lower-Dimensional Processor Arrays, IEEE Trans. on Parallel and Distributed Systems, 7 (4)1996
- [17] V. K. Prasanna Kumar et al., Designing linear systolic array, J. Parallel and Distrib. Computing, vol. 7, 1989
- [18] A. Darte, Regular partitioning for synthesizing fixed-size systolic array, J. VLSI Integration, vol. 12, 1991
- [19] K. Ganapathy, et al., Design a scalable processor array for recurrent computations, IEEE Trans. on Parallel and Distributed Systems, 8(8)1997
- [20] A. L. Fisher et al., Synchronizing large VLSI processor arrays, IEEE Trans. Computers, C-34(8) 1985
- [21] J. Teich, et al., Scheduling of partitioned regular algorithms on processor arrays with constrained resources, ASAP'96, 1996

Web 站点

维护

结构分析法

(18)

图论

计算机科学 1998 Vol. 25 No. 3

73-75, 69

## Web 站点设计与维护中的结构分析法

The Design and Maintenance of Website with Structure Analysis

王宽全

(西南师范大学计算机科学系 重庆400715)

TP393

**摘要** In this paper the structure and problems of Website designed with the traditional approach are analyzed. The structure model of Website and diagnose algorithm are present. At last a new technique of Website design and tools are introduced.

**关键词** Website, Structure, Diagnose algorithm, Visualization tools

## 1 引言

Web 从1991年诞生以来,人们把很大精力投入到漂亮的主页设计上,过分注重了 Web 站点的外在特性,而忽视了 Web 站点的内在结构,随着 Web 站点服役时间的推移,使得 Web 管理者(Webmaster)对 Web 须进行烦琐的维护。由于对 Web 修改越频繁,Web 的页面越多,而且对 Web 站点的内在结构缺乏有力的分析手段,所以对 Web 的维护变得越来越困难。

Jay M. Lightfoot 使用一种基于超文本交互式可视化工具来设计 Web<sup>[1]</sup>,其显然的优点就是将 Web 站点的内部结构用可视化的方法显示出来,对 Web 管理者来说不失为一个维护指南。但他使用的工具 NoteCard 并非专为 Web 设计而作,尽管可以提高 Web 设计与维护能力,却缺乏对 Web 站点结构的自动分析与诊断问题的能力。此外, Bertin<sup>[2]</sup>引入了手工操作网络显示以达到更具结构化设计思想, Eick 和 Wills<sup>[3]</sup>完成了用于设计的抽象网络显示的研究,这些和 Jay M. Lightfoot 使用的 NoteCard 一样可提高 Web 设计与维护的能力。

本文先分析传统 Web 站点设计方法的结构以及存在的问题,再给出 Web 站点的图论模型及问题诊断算法,最后提出一种包含诊断算法的可视化工具。用此工具不但可以加快 Web 站点的设计,而且为日后 Web 管理者对 Web 站点的维护提供极有价值的帮助。

## 2 传统的 Web 站点设计

## 2.1 传统 Web 站点设计方法结构分析

Web 从诞生到现在也只有短短的几年,Web 站点的设计也是一项极新的技术,很难用“传统”来标

志它,然而,对近年来 Web 站点设计的方法作进一步考察,不难看出它们的设计具有相似的过程。为了将它与本文后面要介绍的方法相区别,笔者称它为传统设计方法。传统的 Web 站点设计一般经过以下三个阶段:

第一阶段: Web 站点的初始设计。·设计 Web 站点的主页(Hompage);·设计 Web 站点的其他页面;·建立主页到其他 Web 页面以及其他 Web 页面之间的连接。一般来说,第一阶段的结果是一棵树结构,如图1所示,根结点为主页,是导航的入口,其他分支意味着信息的分类。树中的结点即为 Web 页面,页面中可包含文本、图象、数据文件、Java 的 Applet 等各种信息媒体,一个结点可包含任意个指向其他结点的链,

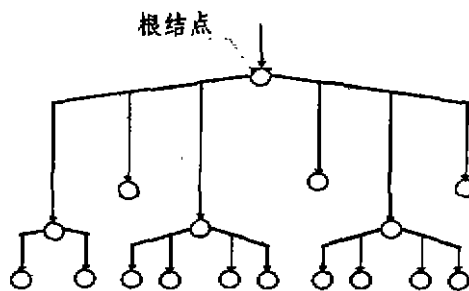


图1 第一阶段 Web 站点内部树结构

第二阶段:在树中加入额外的链。纯树结构只能提供有限的数据库访问能力,没有充分发挥 Web 超文本的全部潜能,因此在 Web 站点内部结点间进行局部互连,使树结构变成网结构,如图2所示。使用网结构设计的 Web 站点提供了最大可能的导航能力。在 Internet 上许多著名站点,如 Yahoo, Microsoft 等它

们都有网结构,而非单一的树结构。

树结构是一个层次结构,对信息具有分类能力,这与现实中的组织机构相对应,不管是对 Web 站点的设计者还是对未来的用户都容易理解,然而,网络虽然增强了导航能力,却模糊了各结点的层次关系,这就是从树结构变为网结构所带来的副作用。

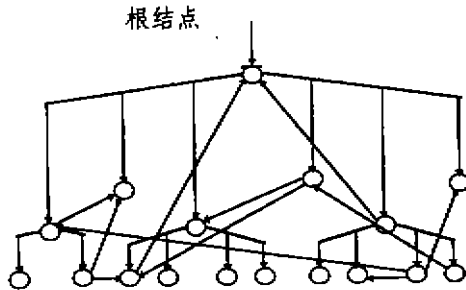


图2 第二段 Web 站点内部网结构

第三阶段:Web 站点维护阶段。完成前两阶段后,Web 站点投入使用,这时开始进入 Web 站点的维护阶段。一段时间后,随着一些新的需求的出现,Web 管理者必须对 Web 作出修改,包括以下内容:加入新页面和链;删除某些页面和链;更新某些页面和链。显然,这些修改必须非常仔细,否则就会产生一些问题。

### 2.2 传统 Web 站点设计方法存在的问题

使用传统方法设计的 Web 站点主要存在以下两方面的问题。从 Web 管理者的角度来看,由于没有一个 Web 站点的结构图来显示结点和相关的链,要确定和更正问题是很困难的。因此,Web 站点的维护变成了一个烦琐和辛苦的事情。这将耗费 Web 管理者大量宝贵的时间,而不能把精力放在更重要的对 Web 站点的信息管理上面。从用户角度来看,这些 Web 站点展示了不合逻辑的结构,很难导航。清晰的树结构不存在了,根结点下面的分支的分类能力薄弱了,因为用户更没有站点的全貌图,当试图查找信息时变得困惑和失意。这将带来对公司站点甚至对整个公司的坏印象,显然这有悖于建立 Web 站点的初衷。这些就是使用传统方法建立 Web 站点的不幸之处。下面通过图3简述设计和维护过程中的结构问题:

·孤立结点(Orphaned nodes)。当指向该 Web 页面仅有的一条链被删除,导致无法链入的孤立结点。

·空链(Dangling Link)。当某链指向的 Web 页面被删除,这时用户沿此链访问将返回一个目标结

点不存在的错误信息。

·孤立子树。当指向某子树的唯一链被删除,导致该子树无法访问。

·环。当两个结点之间存在互为前后链时。

·多结点环。多个结点的链形成闭环。

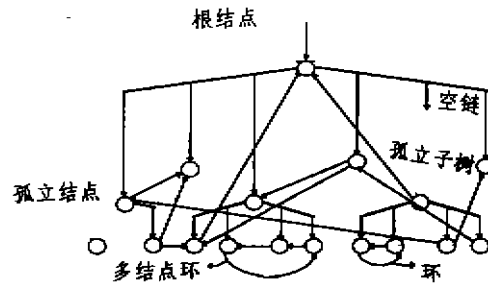


图3 Web 站点内部结构所存在的问题

以上问题在图3中一一标明,不过这些问题不是一朝形成的,而是由于疏于维护积累起来的。不幸的是,由于结构的不规则积累,Web 管理者的维护任务逐渐增加,而问题会逐渐恶化,这就说明了引出基于可视化工具的 Web 设计新技术的必要性。

## 3 Web 的图论模型及问题的诊断

### 3.1 Web 的图论模型

定义 有向图  $D$  是一个三元组  $(V(D), A(D), \psi_D)$ , 其中  $V(D)$  是一个非空的顶点集合,  $A(D)$  是连接  $V(D)$  中的某些顶点对的弧的集合,  $\psi_D$  是关联函数,其定义如下:

设  $a \in A(D)$ , 则  $\psi_D(a) = (u, v)$ , 它是一个有序对偶,称  $a$  是从顶点  $u$  到顶点  $v$  的弧,  $u$  叫做弧  $a$  的尾,  $v$  叫做弧  $a$  的头<sup>[4]</sup>。

一个 Web 站点,如图1、图2所示,我们将 Web 页面作为有向图  $D$  的顶点集合  $V(D)$ , 显然由于主页的存在,它是一个非空集合;从一个 Web 页面指向另一个 Web 页面的链,可以对应于有向图的弧集  $A(D)$ ;有了弧集  $A(D)$ , 其关联函数  $\psi_D$  是显然的,因此一个 Web 站点可以抽象为一个有向图。

一个有向图可以用一个矩阵来表示,设有向图  $D$  有  $n$  个顶点,那么可以用一个  $n \times n$  的矩阵  $R$  来表示。那么矩阵元素  $r_{ij}$  为顶点  $v_i$  到顶点  $v_j$  的弧的条数。

例如有向图  $D$ ,如图4(a)所示,则对应的矩阵表示如图4(b)所示。

从有向图的矩阵表示可以看出:

·一个顶点到另一个顶点无弧,则对应的矩阵元素值为0

•一个顶点到另一个顶点有一条弧,则对应的矩阵元素值为1,否则其矩阵元素值大于1

•某顶点  $v_i$  有自环,则对应的矩阵元素值  $r_{ii}$  等于自环的个数。

从一个实际的 Web 站点来看,上面的三种情况都可能存在,从一个 Web 页面到另一个 Web 页面,可能没有任何连接,可能有一条链连接,也可能有多条链连接。在一个 Web 页面内,也可能有一条或多条链连接,称为页内跳转。所以 Web 站点的结构是一个一般的有向图,而不是一个简单有向图。

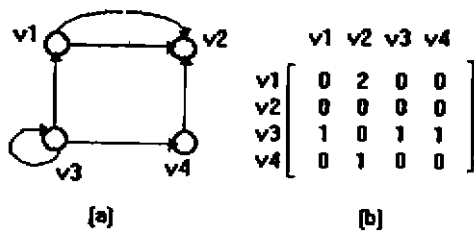


图4 有向图 D 及其矩阵 R

为了有效地诊断前面提出的几个问题,不妨将 Web 站点的一般有向图简化为一个简单有向图,即:

•如果两结点间有多条同方向的弧,则用一条弧代替;

•如果某结点存在自环,则去掉自环。

经上述处理后的有向图为一简单有向图,不存在多条同方向的弧,也不存在自环,在其对应的矩阵中,矩阵元素值仅取0和1,而且对角线元素  $r_{ii}$  总为0。可以证明,简化后的有向图在对 Web 站点结构问题诊断方面与简化前的有向图是完全等价,这里简化有向图的目的只是为了简化诊断算法。

### 3.2 问题诊断算法

主页结点为 Web 站点的入口,因此它不存在问题,下面的诊断算法不包括主页结点,设 Web 站点的结构为简化有向图 D,对应的表示矩阵为 R,诊断算法如下:

3.2.1 孤立结点的诊断,根据有向图 D 和矩阵 R 的定义,有结论:如果  $v_k$  为孤立结点,当且仅当  $r_{ki} = 0, i = 1 \dots n$ 。矩阵 R 中第 k 列元素全为0,也就是没有任何结点有弧链入该结点,在对应的 Web 站点中,无法通过正常导航从主页进入到该孤立页面。

算法:设 G 为孤立结点集, S 为待处理结点集,设  $v_1$  为根结点。

a) 初态:  $G = \emptyset, S = V - \{v_1\}$

b) If  $S = \emptyset$  Then 转 d). Else 取  $v_k \in S, \text{If } r_{ki} = 0,$

$i = 1 \dots n$  Then  $G = G \cup \{v_k\}, S = S - \{v_k\}$

c) 转 b)

d) 算法结束, G 为孤立结点集。

3.2.2 孤立子网 从顶点  $v_1$  (Homepage) 出发,沿各链搜索,如果能到达所有顶点,则不存在孤立子网,否则,无法到达的顶点组成孤立子网。

算法:设可到达顶点集为 S,不可到达顶点集为 X

a) 初态:  $S = \{v_1\}, X = V - S$

b) If  $r_{1j} = 0$  Then  $S = S \cup \{v_j\}, X = V - S$

c) 对所有  $v_k \in S, \text{If } r_{ki} \neq 0$  Then  $S = S \cup \{v_i\}, X = V - S$

d) 算法结束, X 中的各顶点为不可达顶点,如果 X 中仅有一个顶点,则为孤立顶点。

3.2.3 循环结点.所谓循环结点就是在有向图中存在一个有向圈。显然,对于有向圈上的任一结点,它的后继结点总是存在的,那么没有后继结点的结点一定不在有向圈上。

在矩阵 R 中,如果  $r_{ij} = 0, j = 1 \dots n$ ,那么  $v_k$  结点没有后继结点,显然它不在任何一个有向圈上。

算法:设 S 为不在有向圈上的结点的集合, X 为在有向圈上的结点的集合。

a) 初态:  $S = \emptyset, X = V$

b) If  $v_k \in X$ , 矩阵元素  $r_{kj} = 0, j = 1 \dots n$ , Then  $S = S \cup \{v_k\}, X = X - \{v_k\}$ ; 且在矩阵中去掉  $v_k$  所在行和列,即去掉  $r_{ki}$  和  $r_{kj}, j = 1 \dots n, i = 1 \dots n$ , 使矩阵 R 从  $n \times n$  矩阵变成  $(n-1) \times (n-1)$  矩阵。

c) 重复 b) 至 X 为  $\emptyset$ , 或在 X 中无法找到一个满足条件的结点为止。

d) 算法结束, 若  $X = \emptyset$ , 则该有向图中无有向圈存在, 否则 X 中的结点即为有向圈上的结点。

3.2.4 空链. 即有弧而无被指向的结点存在, 上面介绍的图论模型无法诊断此问题。但是, 从前面分析可知, 空链是由于删除了结点而未删除指向它的链而形成。所以, 只要在删除结点之前, 先用前面的算法诊断一下该结点是否为孤立结点, 如果是孤立结点删除之就不会引起空链, 否则先删除链入到该结点的所有链, 使其成为孤立结点再删除之。在 Web 站点维护过程中只要遵从这一原则, 就不会出现空链。

## 4 一种 Web 站点设计与维护新技术

为了克服传统 Web 站点设计与维护中由于缺乏结构指南所带来的问题, 本文提出一种利用可视

(下转第69页)