Internet 的信息收集 Agent 及其搜索方法

沈达阳　林作铨　陈智健　(汕头大学计算机科学研究所　515063)

摘要:本文首先介绍了 Internet 上 Agent 技术的重要性和信息结构,接着阐明了 Internet 信息收集 Agent 的功能及其信息搜索算法,并进一步描述其具体实现的体系结构。

关键词:Internet　软件 agent　信息收集　信息搜索

本文提出了一种基于 Agent 的信息处理方法,即 Internet 上的信息收集 Agent(简称 IICA),旨在加速一定范围内的信息搜集和查询的速度,并提供智能化检索手段,以解决这一问题。

一、WWW 上的信息结构

WWW 基本上是由可通过各种协议(主要是 HTTP)获取的数字化文件组成的数据网络,它提供了一种获取 Internet 上不同资源的统一方式。随着各种界面友好的浏览器(如:Netscape Navigator, Microsoft Internet Explorer)的出现,WWW 的信息正在迅猛发展。

WWW 信息都是以某一站点上的某一页面来表示的,页面文本按照 HTML 标记语言的格式来编写。这些页面可以通过 HTML 中的标记和各种多媒体资源构成链接,从而形成超级文本,同时,还可以通过标记链接到本站点或其他站点的页面,从而形成了数据网络。

各种的 WWW 上各种页面资源一般都是通过统一资源定位标记 URL(unified resources locator)来确定,页面中的各种多媒体资源也都是如此,每一个完整的 URL 包含了:

1. 获取该资源的协议
2. 该资源所在的 Internet 网点位置
3. 该资源在该网点的目录位置
4. 该资源的文件名

不过,WWW 上的页面中的 URL 一般都是不完整的,需要根据具体的面页进行调整和恢复。下面是一个简单的 HTML 页面在浏览器上的外观(如图 1):

由于 WWW 上的信息是以 HTML 页面出现,对 WWW 上的信息的分析,主要就是对 HTML 文本进行分析。由 HTML 语言写成的源文本由标记(用"〈〉"括起来的文字)和文本组成,上面页面的 HTML 源文本如图 2:

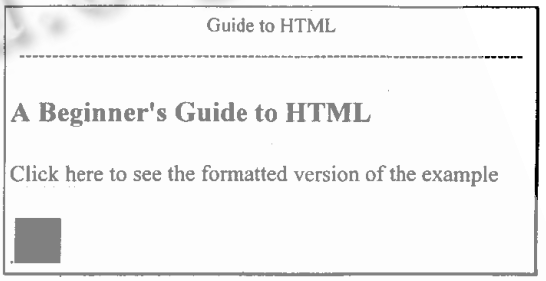

图 1

```
<HTML>
<HEAD>
<TITLE>Guide to HTML</TITLE>
</HEAD>
<BODY>
<H1>A Beginner's Guide to HTML</H1>
<A HREF="MinimalHTML.html">Click here</A>
 to see the formatted version of the example.
<IMG SRC="DoesNotExist.gif">
</BODY>
</HTML>
```

图 2

分析 HTML 时,把由 HTML 语言形成的页面分解成文本块和标记矢量(Tags Vector),而标记矢量中的每一个远素又是一个包含各种属性的复杂特征集(如图 3 所示)。

这些属性大致可以分为两类:

1. 页面中各种资源的 URL
2. 页面元素的形态属性

由于 1,WWW 上的信息才能形成网络体系。因此,

在 WWW 的信息搜索中，主要是通过 1 来实现网点定位的。而通过 2，则可以对页面文本的智能分析，以提供信息搜索所需的启发知识。

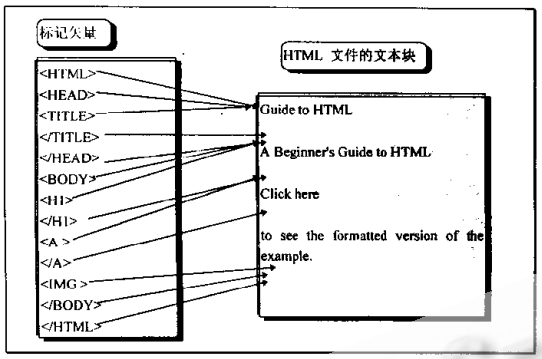

图 3

二、IICA 的功能概述

我们目前所构造的 Internet 信息收集 agent(IICA)是一种 Internet 的软件 agent[1]，它用 Java 编程，可以充分利用 Java 所提供的并发性能。其任务是收集 Internet 上用户感兴趣的信息。它可以采用关键词匹配的方式搜索用户需要的信息，但更重要的是可以自动收集 Internet 上某一个范围内的信息，以数据库的形式保存起来，以加快对该范围内信息的检索速度，IICA 同时根据国内 Internet 的运行特点(如连接国外网点的数据传输率较低，网络联接容易中断等)，提供后台搜索，中断再联等功能。

在 IICA 的服务器端，有一个数据库，它的每一个记录都记录一个 WWW 页面的一些概要信息。该数据库由 IICA 搜索 Agent 自动创建和更新，该 Agent 日以继夜不停地搜索 WWW 上的信息。

在 IICA 的客户端，有另外一个 Agent，它和 Internet 上的用户接触，用户可以输入查询命令，然后该 Agent 启动服务器端的数据管理器，查找用户所要的信息。该 Agent 可以通过 Internet 浏览器启动(http://www.ics.stu.edu.cn/search/)(如图 4 所示)。

IICA 搜索 Agent 可以从一个或多个站点出发，按一定的算法搜索 WWW，并把有关的 WWW 资源(目前可以处理 HTML 文本和其中所链接的图象)，按一定的目录结构保存起来。它还支持对多个站点的同时搜索，也可以同时保存搜索到的多个资源，并有处理搜索过程中出现冲突的能力。

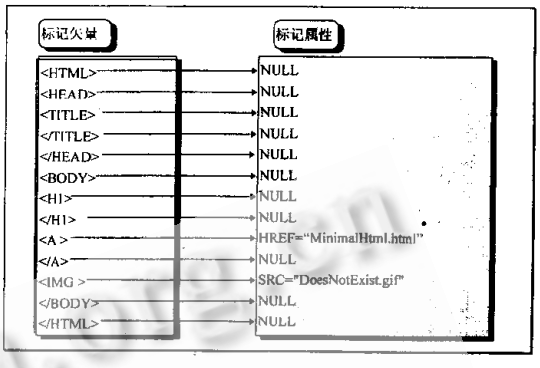

图 4

在信息搜索过程中，IICA 把 WWW 看成是一个由一系列结点形成的无向网，网中的每个结点对应 WWW 上的一个页面。IICA 从若干个结点出发，搜索相邻的结点，并在搜索过程中发现用户需要的内容，在搜索过程中，IICA 采用的是深度优先和有限广度优先相结合的方法：

在一般情况下，IICA 采用深度优先的搜索方法，但当它找到有价值的页面时，如果系统资源允许，它尽可能的采用广度优先的搜索方法，否则，把暂时无法搜索的结点记录在临时性的数据库中。(如图 5 所示)

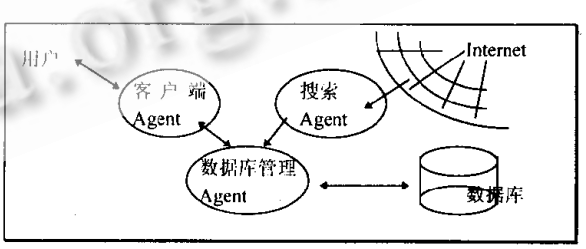

图 5

由于 WWW 是一个巨大的信息库，完全采用广度优先的算法，系统资源的消耗是惊人的。在 Sun Sparc20 (64Mb 内存)，如果以国内的网点为开始结点，一般 10 分钟左右资源就会耗尽，国外的网点由于链接的速度很慢，所以资源耗尽的要慢一些，但搜索到的信息量相同。

三、IICA 所支持的几种搜索算法

1. 主要的数据结构

```
Record of Stack{
    URL url/ * The uniformed resource locator of a document * /
    INDEX index/ * which link is being processed now * /
}
Stack/ * Record the searching process * /
Cache/ * Record the searching files * /
```

2. 算法

(1)深度优先全搜索

```
PROCEDURE FullRecursiveSearch - Depth - First ( startUrl)
{
SetEmpty(stack)
currentUrl←starUrl
currentIndex←null
PushIntoStack(currentUrl, currentIndex, stack)
WHILE IsEmpty(stack) = false DO
{
loop-search:
  WHILE true DO
  {
    currentUrl, currentIndex←GetStackTop(stack)
    IF currentUrl = = null THEN
      GOTO loop-recover
    ELSE{
      htmlFile←Connect(currentUrl)
      PushIntoCache(htmlFile)
      currentIndex←GetALink(htmlFile)
      currentUrl←GetUrl(currentIndex)
      PushIntoStack(currentUrl, currentIndex, stack)
    }
  }
loop-recover:
  WHILE true DO
  {
      currentUrl←PopStack(stack)
    htmlFile←GetFileFromCache()
    IF HasMoreUsefulLink(htmlFile) = true THEN
    {
    currentIndex←GetALink(htmlFile)
    currentUrl←GetUrl(currentIndex)
    GOTO loop-search
    }
  }
}
```

该算法是 IICA 搜索 Agent 最常用的算法

(2)有限区域深度优先搜索

```
PROCEDURE RestrictedSiteSearch - Depth - First ( startUrl)
{
SetEmpty(stack)
currentSiteUrl←GetSiteName(startUrl)
currentUrl←startUrl
currentIndex←null
PushIntoStack(currentUrl, currentIndex, stack)
WHILE IsEmpty(stack) = false DO
{
loop-search:
  WHILE true DO
  {
    currentUrl, currentIndex←GetStackTop(stack)
    IF currentUrl = = null THEN
      GOTO loop-recorer
    ELSE{
      htmlFile←Connect(currentUrl)
      PushIntoCache(htmlFile)
      WHILE true DO / * site restriction here * /
      {
        currentIndex←GetALink(htmlFile)
        currentUrl←GetUrl(currentIndex)
        IF currentUrl∈currentSite THEN
          GOTO loop-find
      }
      PushIntoStack(currentUrl, currentIndex, stack)
    }
  }
loop-revover:
  WHILE true DO
  {
      currentUrl←PopStack(stack)
    htmlFile←GetFileFromCache()
    IF HasMoreUsefullLink(htmlFile) = true THEN
```

```
        }
        currentIndex←GetALink(htmlFile)
        currentUrl←GetUrl(currentLndex)
        GOTO loop-search
        }
    }
}
```

该算法可用于建造某一个 Internet 区域的信息检索数据库。例如，我们可以把搜索的范围限定在“http://www.stu.edu.cn”(汕头大学)的范围内，当搜索过程结束后，一个该范围内所有可以通过页面“http://www.stu.edu.cn/index.html”逐步连接的页面都被登记在数据库中。

(3)宽度优先全搜索(程序清单略)

该算法在资源允许的情况下可以有限使用，以提高搜索的效率。

四、IICA 的体系结构

IICA 的搜索体系由三个层次的 Agent 组成。

·最高层的 agent 是 Monitor，它主要负责系统的监控工作，如系统资源是否耗尽，是否有空闲的 agent 等。

·中层的 agent 是 Collector，它主要负责系统的搜索工作，系统可以同时运行若干个 Collector，在不同的网点进行搜索。

·底层的 agent 是 Analyser 和 Filesaver，它们分别负责系统的 HTML 文本分析和文件存储。

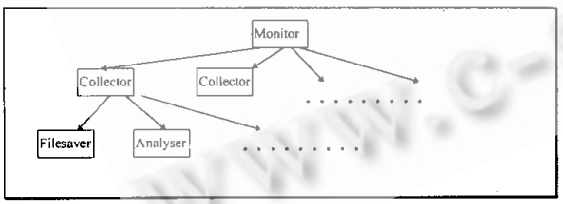

图 6

五、IICA 的设计环境和系统要求

目前 IICA 是以 Sun－Sparc20/solaris2.4/Openwin 为设计环境的，用 Java 语言开发。IICA 在系统中的位置如图 7 所示：

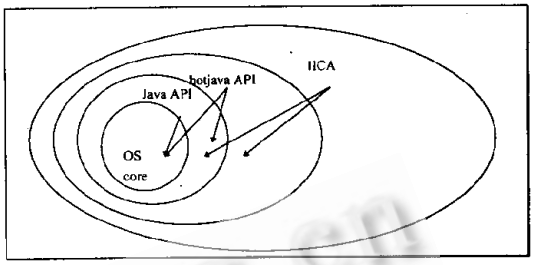

图 7

主要参考文献

[1] 沈达阳，林作铨，Internet 上的软件 Agent，计算机科学，1997 NO.4.

[2] Michael Wooldridge, Nicholas R. Jennings, Intelligent Agents: Theory and Practice, Knowledge Engineering Review, 10(2), June 1995.

[3] 姚郑，高文，软件 agent，计算机科学，23(1)，Jan，1996.

(来稿时间：1997 年 12 月)