

# 基于 PDA 的 Composite 模式及文件搜索算法<sup>\*</sup>

彭红毅, 朱思铭

(中山大学数学与计算科学学院, 广东 广州 510275)

**摘要:** 介绍了 Composite 模式和 PDA 的特点及发展前景, 研究了文件搜索与 Composite 模式的对应关系, 提出了基于 PDA 的文件搜索算法, 该算法能简化客户代码, 并有通用性强, 执行效率高的特点。最后根据该算法对 PDA 的文件搜索应用程序进行了设计, 运行结果验证了该算法的正确性与合理性。

**关键词:** PDA (personal digital assistant); Composite 模式; 文件搜索算法

**中图分类号:** TP391.3 **文献标识码:** A **文章编号:** 0529-6579 (2005) 03-0005-04

设计模式包括 4 个基本要素: 模式名称、问题、解决方案、效果。文 [1] 描述了 23 个模式, Composite 模式是其中的一种。Composite 模式意图是将对象组合成树形结构以表示“部分—整体”的层次结构。文 [2—5] 对 Composite 模式设计进行了相关研究。Composite 使得单个对象和组合对象的使用具有一致性, 描述了如何使用递归组合, 使得用户不必对这些类进行区分, 其关键是一个抽象类, 它既可以表示图元, 又可代表图元的容器。几乎所有面向对象的系统中都有 Composite 模式的应用实例。在 Smalltalk 的 MVC (Model/View/Controller) 结构中<sup>[6]</sup>, 原始 View 就是一个 Composite 模式, RTL Smalltalk 编译框架<sup>[7]</sup> 也使用了大量 Composite 模式。PDA 由于具有体积小, 携带方便, 通讯功能强, 人机界面良好, 信息处理功能丰富, 可长时间使用电池供电, 价格低廉等优点, 其市场前景广阔, 为开发工作留下很大的发挥空间。其操作系统并未统一, 现在 PDA 主要的操作系统有 Newton intelligenc、CEOS、Windows CE、DOS, 其中以 Windows CE 前景最为看好。在 Windows CE 系统下, 其主要开发工具有 EVC (eMbedded Visual C++) 和 EVB (eMbedded Visual Basic)。文 [8, 9] 对 PDA 的设计标准和技巧进行了有益的探讨, 如果将设计模式应用于 PDA 的软体开发, 就会节约大量的时间和精力, 加快项目的进程。但是 PDA 具有体积小等特点, 决定了其开发会受到一些相应限制。例如 PC 机下的 Visual C++ 中能使用的一些函数, 在 EVC 中不能使用, 需要开发人员做相应的开发工作。虽然原 EVC 中有文件搜索程式, 但

是该程式不具有接口形式, 很难移植。为克服此缺点, 本文在此基础上进行了改进, 利用 Composite 模式进行软件设计, 并提出了相应的文件搜索算法。文件搜索算法在 PDA 打开文件、资料备份等软件中都十分有用。该算法具有很强的通用性, 既可用于 MFC, 也可用于 SDK。而且此文件搜索算法也适用于 PC 机下的文件搜索。文章使用 Composite 模式进行程式设计, 从而使代码变得简洁、可移植性强、易于理解。

## 1 基于 PDA 的 Composite 模式及文件搜索算法

在 PDA 上文件都存放在 RAM 中, 放在根目录“\”下 (有些 PDA 为了直观起见, 显示为“My Device”)。如果我们将每个文件夹看成是 Composite 模式中的一个 Component, 文件看成是 Leaf 节点, 那么 PDA 上 RAM 中的文件就是一个树形结构, 且与 Composite 模式具有一一对应关系。文件夹对应 Component, 文件对应 Leaf。文件组合对象结构如图 1, 典型的 Composite 对象结构如图 2。

我们可为文件组合对象声明一个接口, 用于访问和管理文件组合中的组件, 如果搜索到文件, 直接对该文件进行相应的操作; 如果搜索到文件夹, 在文件组合接口中实现与文件组合子部件相关的操作。用户可使用文件组合接口与组合结构中的对象进行交互。

基于 PDA 的文件搜索具体算法设计如下:

步骤 1: 初始化变量 FileFullDirectory (文件路径 + “\”), SubFileFullDirectory (当文件搜索到文

\* 收稿日期: 2004-06-14

基金项目: 国家自然科学基金资助项目 (10371135)

作者简介: 彭红毅 (1973 年生), 男, 博士研究生; 通讯联系人: 朱思铭; E-mail: penghui@yahoo.com.cn

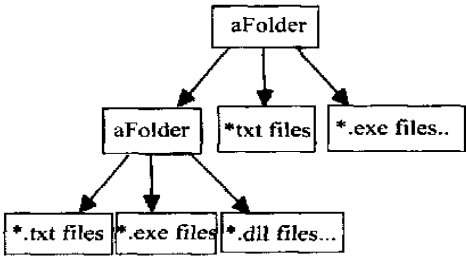


图 1 文件组合对象结构  
Fig 1 File composite object structure

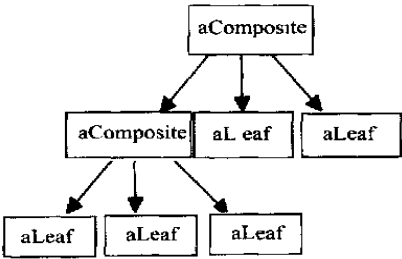


图 2 Composite 对象结构  
Fig 2 Composite object structure

文件夹时，加上此文件夹的新的文件路径)，FileFullName（文件路径 + “ \” + 文件名），FileFullDirectoryAllType（文件路径 + “ \” + “ \*. \*”），nFileCount（搜索到匹配的文件数目），声明变量 fileHandle（文件句柄），findData（找到的文件（包括文件夹）属性结构体）。

步骤 2：在应用程序中调用函数 GetAllDirectoryConents（CString pszDirectory，CString mFileType），确定要搜索的文件夹及文件类型，初始化其中的参数，其中参数 pszDirectory 表示要搜索的文件路径，mFileType 表示要搜索的文件类型。

步骤 3：若文件目录不存在，则输出：“文件目录不存在，请输入正确的目录，并退出！”。

步骤 4：如果 pszDirectory 为 “ \”，则 FileFullDirectory = pszDirectory，否则 FileFullDirectory = pszDirectory + “ \”。

步骤 5：执行语句 FileFullDirectoryAllType = FileFullDirectory + “ \*. \*”。

步骤 6：执行语句 fileHandle = FindFirstFile（FileFullDirectoryAllType，&findData），如果 fileHandle 为无效句柄值，返回。

步骤 7：执行语句 FileFullName = FileFullDirectory + findData.cFileName。

步骤 8：如果 findData.cFileName 是文件夹，转步骤 9；否则，如果 findData.cFileName 是匹配

的文件，获取该文件信息，同时文件数加 1；否则如果是不匹配的文件，将其滤掉。

步骤 9：若 findData.cFileName 是文件夹，获取该文件夹信息，并执行下面语句：

SubFileFullDirectory = FileFullDirectory + findData.cFileName；pszDirectory = SubFileFullDirectory 并转步骤 2。

步骤 10：当 FindNextFile（fileHandle，&findData）为真时，转步骤 7；否则转下一步。

步骤 11：如果文件数为 0，说明没有找到相匹配的文件，则输出“没有找到合乎条件的文件！”。

步骤 12：搜索完毕，结束。

该算法的时间复杂度为  $O(n)$ ，其中  $n$  为文件数目，执行效率高，客户可以一致地使用文件组合结构和单个对象，简化客户代码。若函数 GetAllDirectoryConents 中初始参数为 “ \” 与 “ \*. \*”，则搜索 RAM 中所有类型的文件。如果要在由组合控件、列表控件组成的 Dialog 打开文件的应用程序中使用此算法，只需将步骤 9 修改为：“若 findData.cFileName 是文件夹，获取该文件夹信息”。

2 基于 PDA 的文件搜索程序的设计

第一：声明一个文件物件类，设为 Class CSFileObject: public CObject，这个类包括 3 个元素，1 个操作（或函数）。

元素 1：设为 CString m\_filepath，用来保存文件的路径。

元素 2：设为 CString m\_filename，用来保存文件名。

元素 3：设为 BOOL m\_flag，用来判断获取的文件名是否为文件夹的标志，若 m\_flag 为真，则获取的文件名为文件夹，否则为文件。

操作 1：设为 GetFileInfo（），用来获取文件信息。其代码表示如下：

```
CSFileObject:: GetFileInfo（CString filepath,
CString filename, BOOL flag）
{m_filepath= filepath;
m_filename= filename;
m_flag= flag; }
```

第二：声明一个文件搜索类，不妨为 Class CSearchFiles。这个类包括 8 个元素，1 个操作（或函数）。

元素 1：设为 CSFileObject m\_pFileObject，用来

保存获取的文件信息的物件。元素 2、元素 3、元素 4、元素 5、元素 6、元素 7、元素 8 分别设为 CString FileFullDirectory、CString SubFileFullDirectory、CString FileFullName、CString FileFullDirectoryAllType、WIN32\_FIND\_DATA findData、HANDLE fileHandle、int nFileCount，其含义与基于 PDA 的文件搜索算法步骤 1 中说明的同名变量相同。

操作 1：设为 GetAllDirectoryContents（/\* 文件路径 \*/，/\* 文件类型 \*/），用来获取文件信息。其关键代码表示如下：

```
CSearchFiles:: GetAllDirectoryContents ( CString
pszDirectory, CString mFileType)
{
    .....
    fileHandle          =          FindFirstFile
(FileFullDirectoryAllType, &findData);
    ...
    do { FileFullName= FileFullDirectory + findData.
cFileName; /*在此判定文件类型: */
/*1, 若是文件夹，将文件夹信息加进物件; */
if ( findData. dwFileAttributes & FILE -
ATTRIBUTE DIRECTORY)
{
    ..... SubFileFullDirectory = FileFullDirectory +
findData. cFileName; GetAllDirectoryContents
(SubFileFullDirectory, mFileType);}
/*2, 若是匹配的文件，将文件信息加进物件，同时文件数加 1。 */
.....
/*3, 若是不匹配的文件，将其滤掉。 */
.....}
while (FindNextFile (fileHandle, &findData) ==
=TRUE);
.....}
```

第三：在应用程序中调用文件搜索操作，并提取文件信息。设应用程序类为 Class CDirFilesDlg：public CDialog，这个类包括 3 个元素，1 个操作（或函数）。

元素 1：设为 CString m filepath，用来保存文件的路径。

元素 2：设为 CString m filename，用来保存文件名。

元素 3：设为 BOOL m flag，用来判断获取的文件名是否为文件夹的标志，若 m flag 为真，则获取的文件名为文件夹，否则为文件。

元素 4：设为 int m FileCount，用来记录搜索到相匹配的文件数目。

操作 1，设为 ShowFiles（），用来调用文件搜

索操作，并获取文件信息等。其关键代码表示如下：

```
CDirFilesDlg : ShowFiles ()
{
    CSearchFiles * pDirFiles= new CSearchFiles;
    pDirFiles -> GetAllDirectoryContents ( " \", "
*. *"); POSITION pos = pDirFiles -> m -
pFileObject . GetHeadPosition ();
while (pos != NULL)
{
    CFileObject * pFileInfo= (CFileObject *)
pDirFiles-> m pFileObject. GetNext (pos);
/*取出 pFileInfo-> m filepath; pFileInfo->
m filename ; pFileInfo-> m flag. */
    .....}
    .....}
```

其物件导向（Object Oriented）如图 3。

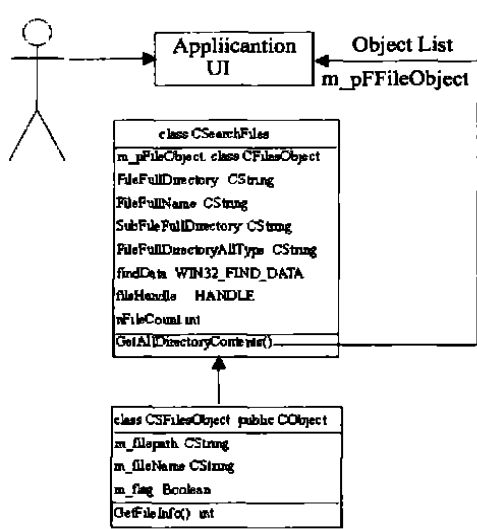


图 3 应用程序的物件导向图

Fig 3 Object Oriented diagram of application

3 实验结果

为了实现本文提出的基于 PDA 的文件搜索算法，我们开发了一个 PDA 的列出文件的应用软件 DirFiles，实验所用设备为一台 PC 机，软件环境为 Windows CE 3.0，Pocket PC，Windows 2000 或更高版本，开发工具为 eMbedded Visual C++。通过这个应用程序运行结果，说明了本文提出的基于 PDA 的文件搜索算法完全能搜索出客户所要的文件，并且在容量和速度都有限的 PDA 上运行效果良好。图 4 为在应用程序 DirFiles 中，对搜索算法初始化参数为“\”和“\*. \*”时的输出显示结果。

4 小 结

本文使用 Composite 模式，提出的基于 PDA 文



图 4 应用程序运行结果

Fig 4 Operating results of an application

件搜索算法具有以下效果:

(1) 定义了包含基本对象和组合对象的层次结构。文件夹基本对象可以被组合成更加复杂的文件组合对象, 而这个文件组合对象又可以被组合, 不断地递归下去。

(2) 简化了客户代码。客户可以一致地使用文件组合结构和单个对象。用户不需要知道处理的是一个文件叶节点还是一个文件夹组合组件。

(3) 使设计更加一般化。使用 Composite 模式进行基于 PDA 的文件搜索, 在运行时刻检查文件组合中的组件。

(4) 最大化文件组合接口。让文件和文件夹尽可能享用公共操作。

## 参考文献:

- [1] GAMMA E, HELM R, VLISSIDES J. Design patterns elements of reusable object-oriented software [M]. New York: Addison-Wesley, 1995.
- [2] YACOB S, AMMAR H H, MILI A. Constructional design patterns as reusable components [J]. Lecture Notes in Computer Science, 2000, 1844: 369—387.
- [3] FRIEDRICH M, HOLGER P. Efficient Object-Oriented Software with Design Patterns [J]. Lecture Notes in Computer Science, 2000, 1799: 79—90.
- [4] FLOCK J, BRAK R. Using SDL for Modeling Behavior Composition [J]. Lecture Notes in Computer Science, 2003, 2708: 36—54.
- [5] SEITER L, MEZINI M, LIEBERHERR K. Dynamic Component Gluing [J]. Lecture Notes in Computer Science, 2000, 1799: 134—146.
- [6] GLENN E, KRASNER T. A cookbook for using the model-view controller user interface paraface paradigm in Smalltalk—80 [J]. Journal of Object-Oriented Programming, 1998, 1(3): 26—49.
- [7] RALPH E, JOHNSON C M, LAKE J M. Code Generation—Concepts Tools Techniques [M]. Dagstuhl: Springer-Verlag, 1992.
- [8] CIAVARELLA C, PATERNO F. Design Criteria for Location-Aware, Indoor, PDA Applications [J]. Lecture Notes in Computer Science, 2003, 2795: 131—144.
- [9] GONZALEZ—CASTANO F J, ANIDO—RIFON L, COSTA—MONTENEGRO E. A New Transcoding Technique for PDA Browsers Based on Content Hierarchy [J]. Lecture Notes in Computer Science, 2002, 2411: 148—158.

## PDA-oriented Composite Pattern and File Search Algorithm

PENG Hong yi, ZHU Si ming

(Department of Mathematics, Sun Yat-sen University, Guangzhou 510275, China)

**Abstract:** Composite pattern, characteristics and foreground of PDA (personal digital assistant) in brief are introduced. The correspondences relation between file search and composite pattern is studied. A PDA-oriented file search algorithm is proposed, which can simplify the client code and is adaptive and efficient. The PDA-oriented file search application according to the algorithm is designed, and the operating results validate its correctness and reasonableness.

**Key words:** PDA (personal digital assistant); Composite Pattern; file search algorithm