*The Design and Implementation of Android Database Service Framework based on Adapter Pattern**

Yiliang Xing
Software Engineering Department
Hainan College of Software Technology
Qionghai City Hainan Province,China
xylpaper@163.com

Yun Pei
Software Engineering Department
Hainan College of Software Technology
Qionghai City Hainan Province,China
Corresponding author py_yunpei@163.com

***Abstract*—Providing database service to android script app can reduce complexity of data management.This paper designs and implements an android database framework based on the adapter pattern. Interface of the framework provides a unified database programming method for PHP script. Adapter of the framework uses android database native service to provide service for the implementation of this method. Adaptee of this framework is a proxy for android database native service. The core code of the framework is also given in this paper. Practice verifies the adaptation, the loose coupling and the expansibility of the framework.**

Keywords—database; android; php; framework; adapter pattern

I. INTRODUCTION

Open source PFA (PHP for Android) [1] is a PHP interpreter running on Android system, which allows PHP applications running on the Android system. Although PFA supports PHP basic syntax, but the PHP application can not directly access Android native services, it needs to access the services through SL4A(Scripting Layer for Android) [2][3]component. SL4A is a software system for the acquisition, development and maintenance of Google Corporation. SL4A provides Android native services for PHP , javascript , python and so on , which include telephone, voice, sms, wifi, intentions services, ect . SQLite[4] is Android native database service, but because the SL4A does not support the service, and therefore scripts relying on SL4A is severely limited impact in data access. SQLite is an embedded relational database engine, which provides data storage service for mobile devices with very limited computing power. SQLite is different from MySQL and Oracle and other professional database, SQLite database is a file, from the point of view, the operation of the SQLite is just a more convenient file operation. SQLite provides two kinds of data operation mode. One is the command line, SQLite contains a command line called sqlite3, which allows users to manually enter and execute SQL commands for the SQLite database. The other is a class library, which allows the application to be developed by Tcl, PHP, C#, Java and so on to use SQLite database. The former has the advantage that the SQLite data operation can be realized without the use of the third party components, but the problem is that the interface of the command interface is usually not fit for applications of PHP and other languages. The advantages of the latter is with good scalability, suitable for the development of a variety of language, but insufficient is according to different development language transplantation third-party components to the Android platform, and its implementation is difficult. This paper discusses the design and implementation of the SQLite database system based on the design pattern of the adapter, which makes it suitable for PHP application in Android platform. This paper is of great significance to open up Android native database service.

II. DATABASE FRAMEWORK BASED ON ADAPTER PATTERN

A. Adapter Design Pattern

The concept of pattern derived from city planning and architectural design works of the architectural design master Christopher Alexander. After a large number of observations, Alexander found that in a specific building, those outstanding structures have their similarities, which are usually in order to solve the same problem. Alexander defined pattern as a solution to a problem in a certain context. In software development, there is also a situation that similar problems are repeated, and these problems can be solved by the same method. As long as these solutions are summed up to form a "pattern", then in the face of similar problems, we can use the existing "pattern" to solve the problems. Design Pattern was originally introduced into the field of software design by Gamma Erich and Helm Richard and so on[5]. Design pattern is a set of code design experience of classification that is used repeatedly, most people know. Adapter pattern is mainly to solve the the problems that interface of an old system can not meet the needs of a new system, old interface will be needed, through adapter transferred to support new interface. In other words, adapter pattern is mainly used in some of the existing system that can continue to play a role in the service for the new system,but there is inconsistent interface between the old and the new system. Adapter pattern is very useful for the legacy code reuse and libraries migration. Adapter pattern can be defined as that one interface is converted to another interface to meet customer requirements, which are not compatible with each other.

According to the different relationship between the adapter and adaptee, the adapter pattern can be divided into two kinds

of object adapter and class adapter . In the object adapter pattern, the relationship between adapter and adapter is associate relationship. In the class adapter pattern, the relationship between adapter and adapter is inheritance (or implementation) relationship. However, with the implementation of "multi inheritance", the high coupling degree is brought, it is generally not recommended to use the class adapter. Object adapter pattern structure is shown in Fig.1. It shows that the pattern structure contains four roles of Target, Adapter, Client, and Adaptee. Client is the customer, which is the terminal object of consumer services. Target is a abstract target, which maybe be an abstract class or interface or a specific class, to meet needs of the Client. Adapter is a class that can convert one interface to another interface, as a converter, Adaptee and Target are well matched by Adapter, Adapter is core of the adapter pattern. In the object adapter, a target is associated with an Adaptee object by adapter which inherits Targe and tie Adaptee. Adaptee ,which is an adapted role, is a class that defines an existing interface, which is generally a concrete class. Adaptee contains business methods and in some cases there may be no source code for the adapter class.

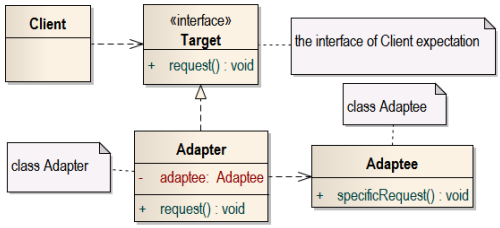

Fig. 1. Object adapter structure diagram

B. Database Framework Design and Implementation

The architecture of the Android SQLite3 service framework based on the adapter pattern and PHP application access SQLite3 database with the framework is shown in Fig.2. It shows that the architecture is composed of PHP Android application, IDB database interface, SQLite3Adapter class and SQLite3Adaptee class. PHP Android application is the database terminal user and it achieves data CRUD(Retrieve, Update, Delete and Create) operation by the IDB database interface. IDB database interface provides unified standard interface PHP Android applications and through this interface PHP Android applications can conveniently realize the data CRUD operations. At the same time, IDB database interface achieves the loose coupling of application and adapter. SQLite3Adapter class has two important role. The first one is to convert the IDB database interface operation to the Android SQLite3 command and realizes the data CRUD operations by the command. This conversion is the key for interface adapter. The second is to convert the operation result of the SQLite3 command to IDB specified data type and so as this simplifies the PHP Android application data programming. Class SQLite3Adaptee agents real SQLite database system. The class contains a method named sqlite3 that docks the sqlite3 command for the database administrator and accepts arguments to do something on database. The method provides 3 parameters, the first parameter represents the database file, the second parameter represents the SQL statement, and the third argument represents the data output file. Default output format of sqlite3 command line is the "list", in list mode, each query record is written in a row and a separator character is separated from the column. By default, the separator character is the pipe symbol "|". List format is shown in Fig.3. It shows that the first line is record fields, which are field1,and field2, other lines are data item values of fileds, the value of field1 are user1, user2, user3, user4 and the values of field2 are pwd1,pwd2,pwd3, pwd4.

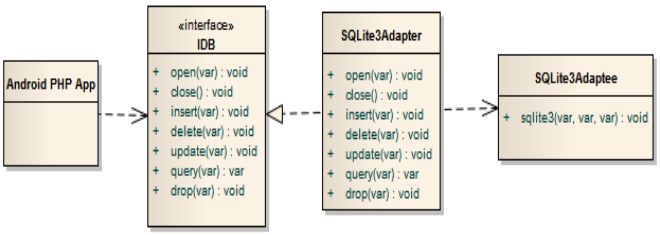

Fig. 2. Architecture of PHP Android App access SQLite3

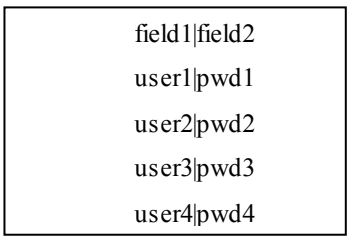

Fig. 3. SQLite3 command output format

Interface IDB defines the CRUD operation. IDB is written as the following.

```
interface IDB
{
    function close();              // close database
    function open($filename);   //open database file
    function drop($query);      //drop database or table
    function update($query);  //update data
    function delete($query);    //delete data
    function insert($query);     //insert data
    function query($query);    // query data
}
```

The aim of Class SQLite3Adaptee is to agent sqlite3 command and assemble sqlite3 command with the parmeters of the database file, SQL statements and output file. The command format is "SQLite3 -header database SQL statement output file", the command will first open the database file, then execute SQL statements in the file to achieve the CRUD operations, finally convert and save the operating results to the output file. The variables of $dbfile, $query and $output in PHP code represent the database file, SQL statements and output files. The core code of SQLite3Adaptee is shown as the following.

```
class SQLite3Adaptee
```

```
{
   function sqlite3( $dbfile, $query , $output)
   {
 $cmd= " sqlite3 -header $dbfile \" ".$query." \" > $output ";
      shell_exec($cmd);
   }
}
```

Interface IDB is implemented by Class SQLite3Adapter, which converts the client interface to the SQLite database system interface. Function named open is to open a database file. If the database file does not exist, it will be created and the default database table named test will be generated. Function named query is to query records of database and store the records to an output file, the output file is the "list" format. To facilitate the application, the list format specification is required to be transformed into a two dimensional array that it uses rows to express records and uses columns to represent fields. The concrete realization process is that firstly the proxy class SQLite3Adaptee will query records of a database and store them to the output file, secondly class SQLite3Adapter will separate the records from the output file in a line and separate fields from the records, finally the separated fields will be heavily composed of a two dimensional array. The functions of insert, update, delete, and drop are to insert, update, delete records of a database and drop a database, the core process to achieve the functions are the same, the process is that the functions will make the three parmeters of database files, SQL statements and output file passed by application layer to a SQL statement and pass the statement to class SQLite3Adaptee to execute. Usually, the functions are used for executing the query on the specified database. Class SQLite3Adapter key code is shown below.

```
class SQLite3Adapter implements IDB
{
   var $SQLite3;                    //SQLite3Adaptee object
   var $filename;                   //Database file
   // SQLite3 query result cache file
var $cachefilename="/mnt/sdcard/sl4a/scripts/cacheQuery.txt";
   function __construct()
   { //Initialize SQLite3Adaptee object
      $this->SQLite3=new SQLite3Adaptee();
   }
   public function open( $file )
   {
      $this->filename=$file;
      if ( !file_exists($file) )
      {    /* if database file does not exist, database file will be created and the default table test will be generated */
         $output=$this->cachefilename;
         $query="create table test('name')";
         $this->SQLite3->sqlite3($file, $query, $cache) ;
      }
   }
   public function close()
   {
      $this->filename="";
   }
   /* Function named query is to query records and $query is SQL statement. Function will return a two-dimensional array. The first record of the array represents fields, other records represents data */
   function query($query)
   {  //1. Firstly execute query commands
      $this->SQLite3->sqlite3($this->filename, $query, $this->cachefilename) ;
      //2. Secondly standardized query results
      //2.1 Read out the query results
      $result=shell_exec("cat $cache ");
      //2.2 Separate the records from results
      $resultArray=explode("\n",$result);
      //2.3 Delete the last record(blank)
      array_pop($resultArray);
      foreach ( $resultArray as $key=>$value ) {
      //2.4 Traversal records
         if ( $key==0 )
         {
      //2.5 If the record is the first record, fields are isolated.
            $columns=explode("|",$value);
            continue;
         }
         else
         {
//2.6 If the record is not the first record, data items are isolated.
            $colDataSet=explode("|",$value);
            foreach($columns as $colKey=>$colValue )
            {  // field=>data item
               $table[$key-1][$colValue]=$colDataSet[$colKey];
            }
         }
```

```
        }
        return $table;
    }
    /* Function named insert is to insert records, $query is SQL statement. */
    function insert($query)
    {
      $this->SQLite3->sqlite3($this->filename, $query, $this->cachefilename) ;
    }
  /* the process of delete, create, update and drop is the same, specific code is omitted. */
  }
```

III. EXPERIMENT AND CONCLUSIONS

The framework of this paper is tested and passed by black box testing and applied in a natural rubber industry economic data system, which is running in the environment of Android+SL4A+PFA+SQLite3. The database of the rubber system contains eight pieces of data table, records of more than 12,000 records. The file size of the database is 806k. Practice shows that the method can effectively solve the interface adaptation problem between application and database. As the method is based on the adapter structure, it can be applied to the database system, and the application code can be reused without modifying the application code. Because the method can change the new function of SQLite3Adaptee, so the method has good expansibility.

The Android database service framework based on adapter pattern can enable PHP applications to access SQLite effectively ,which is Android native database service . It has the characteristics of simple, easy, loose coupling and scalability. The next step work is to extend the framework to a variety of scripting languages.

ACKNOWLEDGMENT

This work was financially supported by the Hainan Natural Science Foundation (20156237).

REFERENCES

[1] irontec. (2015,January). PHP for Android project(PFA)[Online]. Available: http://phpforandroid.net/.

[2] SL4A API Help. (2015,June 10). SL4A API Help[Online]. Available: http://www.mithril.com.au/android/doc/index.html.

[3] P. Ferrill , Pro Android Python with SL4A, Berkeley: APress, 2011.

[4] SQLite. About SQLite. http://www.sqlite.org/about.html.

[5] Erich Gamma, Richard Helm, Ralph Johnson, John Vissides. Design Patterns: Elements of Reusable Object-Oriented Software. New Jersey: Addison-Wesley Publishing Company, Inc. , 2002.