XML and NET Based GUI for Heterogeneous Database Backup System

Tayyeba Minhas, Satish Anamalamudi, Xu Ning, and Minglu Jin

***Abstract*—In order to ensure that data stored on a medium such as a disk is not lost or damaged, a copy of that data is stored on another medium. Should the original data be lost or damaged, then the copy may be accessed to reproduce the original data. This paper explains a novel technique to backup and restore DBMS and all types of files and directory through XML and NET. With proposed Solution, user data will be safely protected from accidental delete, overwrite and viruses. Moreover, proposed methodology is compatible with both oracle and SQL Servers. In this approach, we use XML and Ionic zip file manager to deploy the application. Database structure and metadata of the documents are stored within XML format i.e. all the files and folders are stored using Extensible Mark-up language (XML) instead of data base. With XML, intended data will be created, stored and even transport by defining set of rules with encoding document and proposed algorithm. The main advantage with proposed work is database administrator can use simple XML encoded files to create and backup data from heterogeneous databases (oracle, SQL etc.) at any place instead of using complex tools and interfaces.**

***Index Terms*—Database management system (DBMS), extensible mark-up language (XML), graphical user inter face (GUI).**

I. Introduction

Backup is defined as a copy of data that can reconstruct from database or data-warehouse [1]. Data backup and recovery consists of set of procedures to protect database against data loss and reconstruct the database after any kind of data loss. A relatively simple backup and restore program helps to interface effectively with computer's operating system to backup data including files and directories stored on a hard disk to the tape drive and to restore such data from the tape drive onto the hard disk [2]. Backup can be stored at a specific folder where the file may be either in coded form or in normal form i.e. the application has two kinds of actions namely automatic and manual approach. In first approach, backup will be created automatically on widow's start up i.e. user cannot see interface [3]. At first start up, user needs to define automatic backup option. Subsequently, application will save the settings and data will be automatically gets backup from subsequent backups. In later approach, user should be able to use the application interface to backup at any time in any file or folder or any data base. The main advantage with manual approach is user backup will be more secure compared with automatic approach because application can backup at heterogeneous places instead of one place in automated system. In our approach, both automated and manual approaches are implemented to backup user data. The rest of the paper is organized as follows. Section II describes about related work and pros and cons of existing solutions. Section III explains about functional requirements of application, Section IV explains about proposed system architecture. Section V explains about implementation work and finally ends with conclusion in Section VI.

II. Related Works

In general, three types of backup namely full, differential and transaction log Backups are available. In order to store and recover data, most of the organizations use more than one database management system. Moreover, each of these servers has specific interface and tools to backup and recover the user data. Hence, database administrator has to learn all these tools for different servers which are very difficult and time consuming task [4], [5]. Moreover, existing backup products like ARCserveIT, brightStor and NetBackup are very costly for medium markets to afford for multiple data processing in heterogeneous platforms [1], [6]. Thus, a simple, reliable, robust and cost efficient back system are needed to backup data from multiple sources. This paper proposes XML and .NET GUI based backup and recovery system for heterogeneous database backup systems.

III. Functional Requirement

Application should have the mechanism of authentication. At first start-up of the application, it should ask the user to select authentication mode from the following: Password protected (user must enter user id and password every time to use the application). Default (no password required to use the application)

User should be able to define the physical paths (paths on hard disk) for backup and restore operations.

If the user opt to go for default the default path and default functionalities are used.

User should be able to define the backup overwrite settings, so that the existing backups can be managed without any problem.

Databases can also restore and backup by using the same approach as described in the above points.

Manuscript received December 30, 2012; revised March 22, 2013.

T. Minhas and Xu Ning are with School of Computer Science, Wuhan University of Technology, Wuhan, P.R.China (e-mail: knoor50@yahoo.com, xuning@whut.edu.cn).

S. Anamalamudi and M. L. Jin are with School of Information and Communication Engineering Dalian university of Technology, Dalian, P.R.China (e-mail: satishnaidu80@gmail.com, mljin@dlut.edu.cn).

DOI: 10.7763/IJIET.2013.V3.288

Application shall provide the user with some options (Overwrite existing backup, delete existing backup to recycle bin, delete existing backup permanently)

User should be able to perform backup/restore operations manually and automatically:

For manual operation, user shall be presented with a windows (desktop) application so that the user can perform the backup, restore, change in settings operations manually. For automatic operation user shall save settings for automated backup/restore operations at start-up, or time-to-time backups. For this purpose, the application shall utilize windows start-up, registry operations and windows services.

IV. System Architecture

In this paper, a new technique using Extensible mark-up Language (XML) based GUI which is compatible with all types of servers like SQL Oracle etc. are proposed.

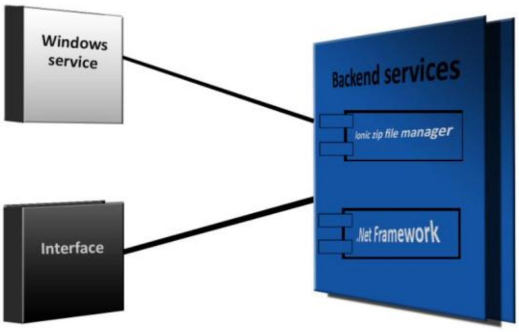


Fig. 1. Backup and restoring service.

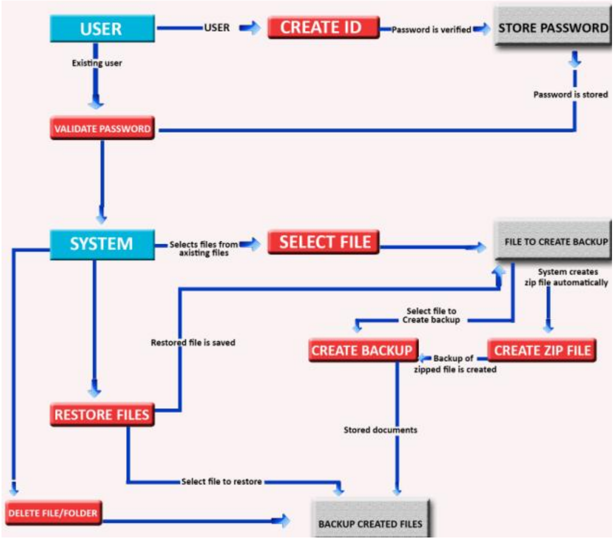


Fig. 2. Data flow diagram for proposed backup and recovery architecture.

With each stored file, we attached a frequency information XML file which allows a solution backup and restore the database by choosing specific articles without learning how does each DBMS work, and also without opening them. The administrator can choose a specific article for Backup/Restore instead of the Backup/Restore the whole DBMS. Fig. 1 explains about backup and restoring window services through compressed form. These services are also used to store real-time information i.e. postfix or prefix is stored in the form of some predefined codes at allocated position of the storage device. The main advantage with this is user can easily understand each backup copy of data with process code number. Fig. 2 explains about data flow diagram for proposed backup and recovery system. Firstly, user authentication and authorization will be checked through credentials like username and password. Secondly, selected backup files will be compressed through ZIP and store in "Create Backup files". Finally, backup files are stored in selected storage backup device.

Fig. 3 shows about proposed three layer architecture for database backup and recovery through XML and .NET GUI. The top most layer of Fig. 3 is named as Access layer.

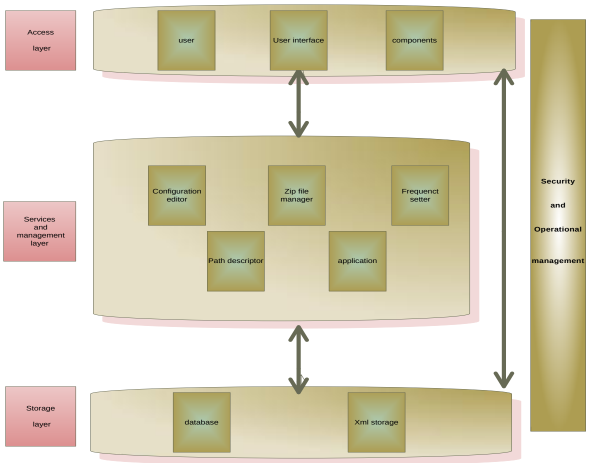
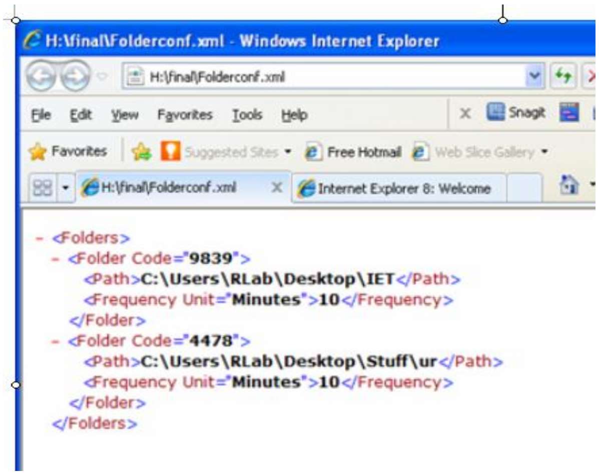


Fig. 3. Three layer architecture of proposed backup and recovery system.

It contains user interface and components subsection. Database administrator can access second and third layers by providing authentication and authorization information. It is crucial to use important cryptographic algorithms to provide better security for authentication, authorization and encryption. X`RSA asymmetric cryptographic algorithm with Diffie–Hellman key exchange is used to provide user authentication and authorization. DES 64-bit symmetric cryptographic algorithm is used at the time compressing user data. Hence, backup data will have safe and secure retrieval.

Service and management functions are explained in layer.2 of Fig. 3. One of the most important beauties of this work is that the data recovery process can not affect the activities in real-time task which will avoid real-time task reach to the transition state. .Net is used to create database tables which can read/write from/to XML form. Using read XML; tables are exported into XML data set without database action. The sample file of XML code is shown Fig. 4 below.

Once path is settled on a backup method, application needs to set up a schedule for running backup. Determining the exact location where the backup data will be stored consists of set of procedure for testing backups to ensure that they can be restored correctly. With proposed algorithm, a method is developed to create a backup and restore the lost data. In order to improve the system performance advance, XML is used to store it in WinZip with advanced symmetric key cryptographic algorithm. One way to encode the path selection process in data backup and recovery application is through path that contains the information about the location where the data backup should be kept after selecting the original path. "DB Function" message will be shown which describes that the backup is successfully stored.

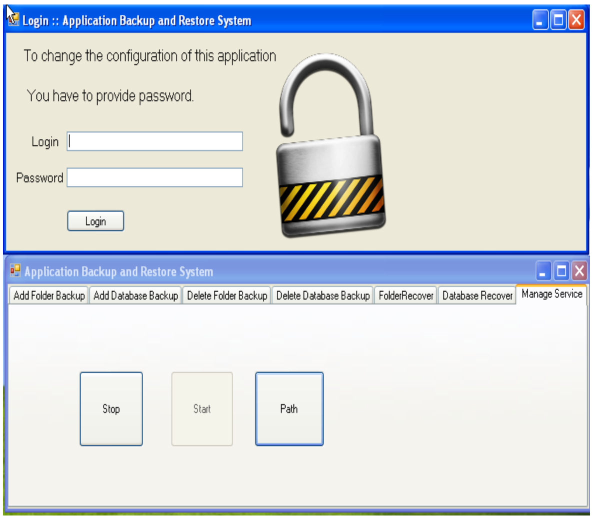

H:\final\Folderconf.xml - Windows Internet Explorer
H:\final\Folderconf.xml
File Edit View Favorites Tools Help
Snagit
Favorites
Suggested Sites
Free Hotmail
Web Slice Gallery
H:\final\Folderconf.xml
Internet Explorer 8: Welcome

```
- <Folders>
  - <Folder Code="9839">
      <Path>C:\Users\RLab\Desktop\IET</Path>
      <Frequency Unit="Minutes">10</Frequency>
    </Folder>
  - <Folder Code="4478">
      <Path>C:\Users\RLab\Desktop\Stuff\ur</Path>
      <Frequency Unit="Minutes">10</Frequency>
    </Folder>
  </Folders>
```

Fig. 4. XML sample code to create and store data.

V. Implementation

Application starts with login and password GUI. Once authentication and authorization is valid user will access the entire application services. Fig. 5 describes about various services that application provides to database administrator once authenticated.



To change the configuration of this application

You have to provide password.

Fig. 5. GUI scenario of proposed backup and recovery application.

Application resides in the system at the time of start-up and creates the backup of selected database files and folders after a prescribed time mentioned by the user. User selects a database or file to create a backup or delete an existing file. “Change path” is used whenever, user needs to create a new path for creating backup or one needs to alter any existing path. Restore data is the option by which user can manually restore file or database. The proposed system is designed in such a way that whenever, the original data on storage disk is accidentally erased, overwritten, or becomes inaccessible because of a hard disk malfunction or sudden power fluctuations the data will be safely retrieved from the “Data Recover” option of Application Backup and Restore System. Along with each backup file an XML file is created that contain data tables of the data set and information about where it is saved.

Whenever, an XML document is created over a relational database system a schema based generation technique is used to automatically create relational tables for storing XML documents. Inserted XML documents are shredded and stored as rows in these tables.

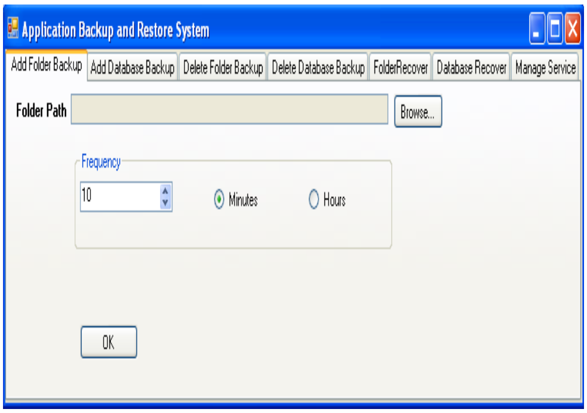


Fig. 6. Backup frequency change duration scenario.

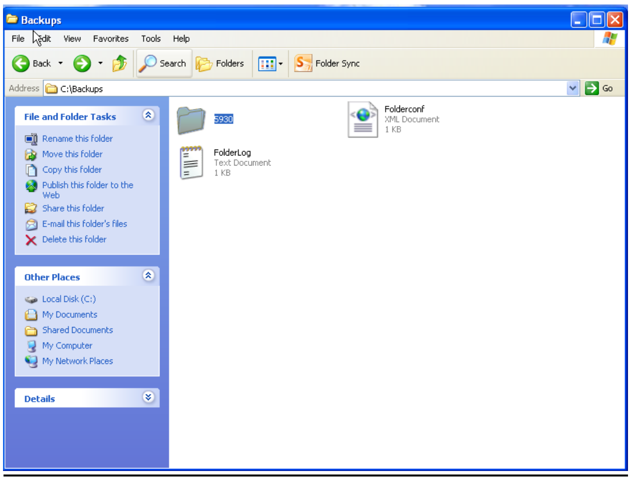


Fig. 7. Data creation and backup scenario in storage disk.

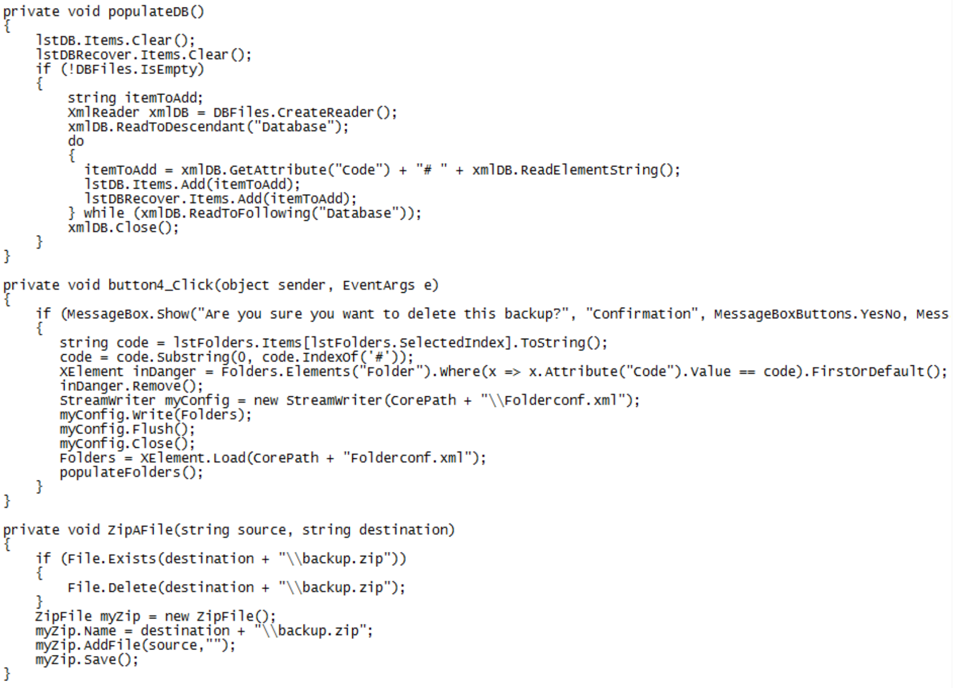
In addition, a reconstruction XML view is created over the created relational tables, which (virtually) reconstructs the “stored” XML documents from the shredded rows. The reconstruction XML view is specified just like a regular XML view of relational data. Queries over the stored XML documents are proposed to enable users to query seamlessly across relational data and XML documents. This provides users with unified access to both relational and XML data. However, using above proposed technique, it is sufficient to just generate a reconstruction XML view instead of writing a full-blown XML query processor. Document-centric information is manipulated in specialized ways to specify their media types. Fig. 6 explains about the backup frequency change scenario option in proposed application. Database administrator can set the duration of time to store data in selected path. Fig. 7 shows the physical location for data creation and backup which is selected by user dynamically for specific duration shown in Fig. 6.

VI. Conclusion

This paper proposes a design of XML-based GUI for heterogeneous database backup system which helps user to create, store and retrieve data from multiple database with simple XML files. Moreover, user doesn’t require a separate tool for each and every database to create and store the data. This application is developed through .Net which store real time information in XML file and attach it along with the file in the system that could help when restoring the file in on demand basis.

APPENDIX

Sample Code



```
private void populateDB()
{
    lstDB.Items.Clear();
    lstDBRecover.Items.Clear();
    if (!DBFiles.IsEmpty)
    {
        string itemToAdd;
        XmlReader xmlDB = DBFiles.CreateReader();
        xmlDB.ReadToDescendant("Database");
        do
        {
          itemToAdd = xmlDB.GetAttribute("Code") + "# " + xmlDB.ReadElementString();
          lstDB.Items.Add(itemToAdd);
          lstDBRecover.Items.Add(itemToAdd);
        } while (xmlDB.ReadToFollowing("Database"));
        xmlDB.Close();
    }
}

private void button4_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Are you sure you want to delete this backup?", "Confirmation", MessageBoxButtons.YesNo, Mess
    {
       string code = lstFolders.Items[lstFolders.SelectedIndex].ToString();
       code = code.Substring(0, code.IndexOf('#'));
       XElement inDanger = Folders.Elements("Folder").Where(x => x.Attribute("Code").Value == code).FirstOrDefault();
       inDanger.Remove();
       StreamWriter myConfig = new StreamWriter(CorePath + "\\Folderconf.xml");
       myConfig.Write(Folders);
       myConfig.Flush();
       myConfig.Close();
       Folders = XElement.Load(CorePath + "Folderconf.xml");
       populateFolders();
    }
}

private void ZipAFile(string source, string destination)
{
    if (File.Exists(destination + "\\backup.zip"))
    {
        File.Delete(destination + "\\backup.zip");
    }
    ZipFile myZip = new ZipFile();
    myZip.Name = destination + "\\backup.zip";
    myZip.AddFile(source,"");
    myZip.Save();
}
```

ACKNOWLEDGMENT

The authors would like to thank for anonymous reviewers for their invaluable comments and suggestions.

REFERENCES

[1] H. J. Chen and Z. Zheng, "Design and Implementation of XML-based GUI for Cross Platform Backup System," in *Proc. International Symposium on Computer Science and Computational Technology*, vol. 2, Dec. 20-22, 2008, pp. 380-382.

[2] X. F. Wang, "XML in the application of network test system," in *Proc. International Conference on Electrical and Control Engineering*, Sept. 16-18, 2011, pp. 937-940.

[3] W. Itani, A. Kayssi, and A. Chehab, "SECERN: A Secure Enterprise Backup and Recovery System for Mission-Critical Relational Database Servers," *Innovations in Information Technology*, pp. 1-5, Nov. 2006.

[4] R. V. Jayakumar *et al.*, "A backup mechanism with concurrency control for multilevel secure distributed database systems," in *Proc. Third International Conference on Digital Information Management*, Nov. 13-16, 2008, pp. 57-62.

[5] K. X. Hu, S. Mehrotra, and S. Kaplan, "Failure handling in an optimized two-safe approach to maintaining primary-backup systems," in *Proc. Seventeenth IEEE Symposium on Reliable Distributed Systems*, Oct. 20-23, 1998, pp. 161-167.

[6] M. Y. Han and Y. L. Yoon, "An implementation and performance analysis of backup system using concurrent log processing in real-time DBMS," in *Proc. Fourth International Workshop on Real-Time Computing Systems and Applications*, Oct. 27-29, 1997, pp. 118-125.

Tayyeba Minhas is currently a PhD candidate in School of Computer Science and Technology, Wuhan University of Technology, Wuhan, China. Her research interests include database and datwarehouse management systems and MAC protocol design in Mobile ad-hoc networks.

Satish Anamalamudi is currently a PhD candidate in School of Information and Communication Engineering, Dalian University of Technology, Dalian, China. His research interests includes Common control channel design for MAC and routing protocols in Cognitive radio ad-hoc networks, Performance enhancement of TCP in mobile and cognitive radio ad-hoc networks.

Ning Xu received his Ph. D. degree in electronic science and technology from the University of Electronic Science and Technology of China in 2003. Later, he was a postdoctoral fellow with Tsinghua University from 2003 to 2005. Currently, he is a professor at the Computer Science Department of Wuhan University of Technology. Dr. Xu's research interests include computer-aided design of VLSI circuits and systems, computer architectures, data mining, and highly combinatorial optimization algorithms. He has published over 50 research papers and led over 5 research projects in these areas. He is a senior member of China Computer Federation and the Chinese Institute of Electronics.

Minglu Jin is a Professor in the School of Electronics & Information Engineering at Dalian University of Technology, china. He received the Ph.D. and M.Sc. degrees from Beihang University,china, the B.Eng. degree from University of Science & Technology of China. He was a Visiting scholar in the Arimoto Lab. at Osaka University, Japan from 1987 to 1988. He was a Research Fellow in Radio and Broadcasting Research Lab. at ETRI, Korea from 2001 to 2004. Professor JIN's research interests are in the general areas of signal processing and communications systems. Specific current interests are nonsinusoidal function theory and its applications, power amplifier linearization, radio over fiber, nulling antenna techniques.