A Mobile Solution for Arboriculture Management: An SQL Server Integration Services Approach

Ntekop, Mfonobong Maurice and Umoh, Uduak Augustine

Department of Computer Science, University of Uyo, Uyo, AKS, Nigeria.

Corresponding Author: Umoh, Uduak Augustine

Abstract
Trees contribute immensely to our environment in areas such as provision of shelter, flood prevention, rainfall dispersal, erosion prevention, climate stabilization global warming, etc. This has increased the involvement of governmental and non-governmental organizations in tree planting. Although tree planting is admirable, lack of good maintenance culture may daunt the expected result of such practice. Currently, there are a variety of software solutions available to help arborist perform their duties efficiently. However, due to technological advancements, these tools are limited in some areas such as robustness. This paper designs, implements and evaluates a mobile solution for arboriculture management using SQL Server Integration Services (SSIS) approach. The SSIS approach is developed using SQL server 2008, C# programming tool, object oriented design method. This work will help easy storage and retrieval of tree information and also enhance good maintenance practice in tree planting.

Keywords: SSIS package, ETL, PDA, database, global warming,

INTRODUCTION
The importance of trees is evident when one envisions a world without them. Typically, trees contribute immensely to our environment in areas such as provision of summer shade and winter shelter, flood prevention, rainfall dispersal, erosion prevention, and climate stabilization. Also, there is a growing global concern about the importance of trees in global warming mitigation. Tree trunks are presented as the largest reservoir of carbon worldwide (Huang et al, 2009). This awareness on the importance of trees has increased the involvement of governmental and non-governmental organizations in tree planting. Although tree planting is admirable, lack of good maintenance culture may daunt the expected result of such practice.

In UK and other parts of the world, local authorities are actively involved in arboriculture. Apart from planting and maintaining council-owned trees, they ensure that privately owned trees receive adequate care from their owners. Owing to the immensity of task and complications that exist in this field, most local authorities have migrated from the traditional approach of maintenance – which involves the use of paper cards to collect and store data – to computerised approach because of the limitations of the former.

There are a variety of solutions including web-based solutions which allow members of the public to log in complaints about trees and track work progress. It also stores tree survey details, inspection details and job details carried out by woodland officers. These have helped local councils and arborist perform their duties efficiently. (Andres, 2004) (Wu et al, 2012) (Fonseca et al, 2012) (Dufour-Kowalski, et al, 2012) (Leskey et al, 2008) (Wang, et al, 2013) (Porte & Bartelink, 2002). However, due to technological advancements, some tools are limited in certain areas. One of such limitations is that arborists will have to enter data into the database manually.

In this paper we design, implement and evaluate a mobile solution for arboriculture management. We adapt and modify the model based on (Andres, 2004) to eliminate the paper process and manual update of the database server without greatly increasing the operating cost. It focuses on investigating ways of enhancing onsite data collection, tree works ordering and budgeting, enquiries and complaints and tree preservation orders for arborist. The proposed approach involves data collection on Excel mobile template and the creation of SSIS package to perform Extract, Transform and Load (ETL) transaction. We explore object oriented database tool in the design of our database and implement in SQL Server Integration Services environment using C++ programming tool. The developed system incrementally loads the database while it loops over excel mobile files transferred from various Personal Device Assistants (PDA) into a shared folder.
Coligny, 2007) (de Coligny, 2005). Contrary to the paper system of collecting survey data available years back, these systems have changed the way arborists perform their duties. As a result, there is a remarkable savings in terms of time and money in the industry. Companies such as EzYTreev, Arbortrack, Woodplan and Robin Forestry Surveys can boast of products that are linked to the Global Positioning System (GPS). The role of GPS is to provide a real time view of trees based on their positioning (longitude and latitude). A GPS-enabled management system provides the arborist not only with the exact position of the tree, but also with the visuals of the tree’s location and condition.

Geographical Information System (GIS) is another technology associated with tree management systems. EzYTreev, Arbortrack, WoodPlan and exegesis have products that link with GIS. Most local authorities take advantage of this since they already have GIS existing in their local councils. By having a GIS link, a management system is endowed with integrated digital mapping, which is used to identify underground utilities that trees may pose a threat to (Wang, et al, 2013) (Porte´ & Bartelink, 2002) (Maastikuarhitektid, 2012).

Tree Management systems are user specific. While there are systems developed for local councils, majority of the systems available are for smaller firms or individuals. Small capacity tree management systems can cater for 10,000 trees approximately and cost about £1,000 pounds while those targeted at the local councils will usually accommodate between 50,000 to 100,000 trees and is estimated for £5,000 (Shamash, 2007). EzYTreev, Arbortrack and WoodPlan companies are Tree Management system industry leaders. They offer systems that run entirely on personal computers and modular systems that have some modules running on Personal Device Assistant (PDA) to enable onsite data collection. The information is usually transferred to workstations/servers manually or through wireless/wired devices remotely or locally (Shamash, 2007) (Fonseca et al, 2012) (Dufour-Kowalski, et al, 2012) (Leskey et al, 2008) (Dreyfus, 2012).

By perusing through the report produced by (Andres, 2004) it is established that his tool addresses three out of the five key features available in today’s Tree Management System (TMS), which include; tree works ordering and budgeting, enquiries and complaints and tree preservation orders. Provision is not made for onsite tree data collection and integrated digital mapping which our paper has filled the gap. Object oriented database model is employed in the design of the database. Information systems developed with the traditional approach have been notoriously error prone, expensive and inflexible. The object oriented approach has the potential to reduce errors, cost and increase flexibility because of its inherent features. (Thalheim, 2000) (Conolly and Begs, 2005) (Ramakrishnan and Gehrke, 2002).

RESEARCH OBJECTIVES
Our main objective is to provide a portable, reliable and affordable tree management tool for small and medium scale enterprises, that is capable of performing ETL transactions and data retrieval by various users at all times.

RESEARCH METHODOLOGY
A Mobile Solution for Arboriculture Management

Database Models: Figures 1 and 2 show the architectures of conceptual and logical database models for arboriculture management. Sample data job table is presented in Table 1.

Conceptual database design stage is the stage where database entities and their attributes are identified, along with potential entity relationships, primary and foreign keys. Sixteen entities are identified and produce in UML as shown in Figure 1. In Figure 2, many-to-many relationship exists between Inspection table and Condition table. In order to resolve this many-to-many relationship, a table is created and name InspCondition. InspCondition is the only weak table in the logical data model. This table has as its attributes two foreign keys, which references the primary keys of Inspection and Condition tables.
Table 1: Job Table

<table>
<thead>
<tr>
<th>JobId</th>
<th>TreeId</th>
<th>LocId</th>
<th>Date</th>
<th>Task</th>
<th>Prior</th>
<th>Duration</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>5</td>
<td>19/04/10</td>
<td>Cleaning out</td>
<td>1</td>
<td>1 days</td>
<td>Pending</td>
</tr>
<tr>
<td>2</td>
<td>78</td>
<td>6</td>
<td>12/12/09</td>
<td>Fell and replace</td>
<td>2</td>
<td>2 days</td>
<td>In progress</td>
</tr>
<tr>
<td>3</td>
<td>321</td>
<td>2</td>
<td>25/05/09</td>
<td>15% Thin crown</td>
<td>3</td>
<td>6 hours</td>
<td>completed</td>
</tr>
<tr>
<td>4</td>
<td>87</td>
<td>4</td>
<td>04/04/09</td>
<td>Tree planting</td>
<td>4</td>
<td>3 hours</td>
<td>Completed</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>23</td>
<td>23/02/10</td>
<td>Remove obstruction</td>
<td>3</td>
<td>2 hours</td>
<td>Suspended</td>
</tr>
<tr>
<td>6</td>
<td>45</td>
<td>11</td>
<td>28/07/08</td>
<td>Weed</td>
<td>2</td>
<td>1 hour</td>
<td>In progress</td>
</tr>
<tr>
<td>7</td>
<td>32</td>
<td>13</td>
<td>09/08/08</td>
<td>Silvicultural thinning</td>
<td>5</td>
<td>3 days</td>
<td>Pending</td>
</tr>
</tbody>
</table>
A Mobile Solution for Arboriculture Management Model: Figure 3 presents SSIS model.

The Figure 3 shows a mobile solution for arboriculture management model. The worksheet is designed to have 4 sheets to accommodate those job functions required for onsite data capture, and a 5th for data validation task. The onsite job functions are: tree survey, tree inspection, job status update and job inspection. Data validation controls the sort of data that can be entered into specified cells of an excel workbook. The PDA is synchronized with the server before files are transferred designated shared folder. A connection manager is used to connect to the data source. Data flow Loop is designed to repeat the data flow task for all excel worksheets present in the shared folder. This looping will be implemented by inserting all data flow tasks in a Foreach loop container in the control flow layout, and its enumerator is configured to loop over excel files in the designated shared folder only. Incremental load is used to load data from a data source to SQL Server. Look-up and conditional split data flow transformations are utilised to implement incremental load. Sequential data loading is employed to ensure that Tree data flow task executes first. All errors are redirected to a flat file destination where such errors are viewed, corrected and inputted into the database.

RESULTS AND DISCUSSION
The TeeDbase is implemented on SQL server 2008 as shown in Figure 4. Microsoft Excel 2003 is employed to implement mobile template which is used by contractors and woodland officers for onsite data collection. Data validation model is designed by setting the data validation value of the affected fields to the primary keys of the referencing tables. Figures 5 shows the original excel connection manager’s connection string used as the variable’s value. The updated connection string, which is an expression is presented in Figure 6.

---

Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\Users\home\Documents\Visual Studio 2008\Projects\Integration Services\Project5\Integration Services\Project5\EXCELBS\Book1.xls;Extended Properties="EXCEL 8.0;HDR=YES";
Fig. 6: Updated Connection string

*Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" + @[User::Filename] + ";Extended Properties=\"EXCEL 8.0;HDR=YES\";*

Figure 7 shows control flow interface indicating all the connection managers, the foreach loop, and a sequential arrangement of the data flow task in the foreach loop according to execution preference. After configuring the dynamic source connection, excel file looping is completed by adding a foreach loop container to the control flow. The container is configured to access the shared folder and loop over .xls files only (which is MS excel 2003 file format). This is preferred because it reduces the complexity of the system. The linked up data flow tasks are then dropped in the foreach container. As a result, the sequence is repeated for each excel file that is found in the shared folder. Figure 8 presents data flow task showing incremental load structure. Figure 9 shows Lookup Transformation Error Output. The lookup transformation gets its output by equi-joining input columns (left) with referenced data set. In order to achieve a left join, so that it returns all rows from the source table. This is configured to ignore failure on matched output.

The conditional split transformation as shown in Figure 10, routes data to diverse outputs depending on the input data content. In this case, an expression was created for 3 different source data types: New, changed and unchanged rows. The condition for new rows is the presence of a non-existing destination database primary key, and that of changed rows is the occurrence of a change in an existing column. The default output is unchanged rows, which flows out of the pipeline. The OLE DB Command Transformation update code as shown in Figure 11, is used to update the destination database with any changed data row that is outputted. Tree survey details are presented in Figure 12. We carry out performance evaluation on the approach based on the following parameters; security, availability/reliability, connectivity to mobile device, connectivity to database, scalability and error handling, database update, usability, interoperability, cost, maintenance and performance. These parameters are ranked based on low, medium and high fuzzy sets. Results indicate high security, high reliability/availability, low connectivity to mobile device, high connectivity to database, low scalability, high error handling, low database update, high interoperability, medium cost, high maintenance and medium performance.
Fig. 9: Lookup Transformation Error Output

<table>
<thead>
<tr>
<th>Order</th>
<th>Output Name</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>New</td>
<td>ISNULL(DTree_Id)</td>
</tr>
<tr>
<td>2</td>
<td>Changed</td>
<td>(ISNULL(DTree_LocId) + &quot; &quot; + DTree_LocId) = ISNULL(DTree_Id) + &quot; &quot; + DTree_Id).</td>
</tr>
</tbody>
</table>

Fig. 10: Conditional Split Transform Editor

```
UPDATE [dbo].[Tree]
SET
```

Fig. 11: Tree table update code in an OLE DB Command Transformation.
CONCLUSION
In this paper, we have presented a mobile solution for arboriculture management (an SSIS package). The application which, adequately load the tree database with tree data captured onsite. We have explored Server Integration Services (SSIS) approach, which collect data on Excel mobile template and perform Extract, Transform and Load (ETL) transactions. The SSIS approach is implemented by creating a package that incrementally loads the database with tree data while it loops over excel mobile files transferred from various Personal Device Assistants (PDA) into a shared folder stored in excel format. Performance evaluation carried out shows that this approach is less susceptible to failure, cheap and is more reliable.

REFERENCES


