Journal of Emerging Trends in Engineering and Applied Sciences (JETEAS) 4(3): 359-366
 (ISSN: 2141-7016)
jeteas.scholarlinkresearch.org

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 no-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 envisages 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 allows 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.

LITERATURE REVIEW

There are a variety of computerised systems available for tree management (*de Coligny* et al, 2010) (*de*

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 carter 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

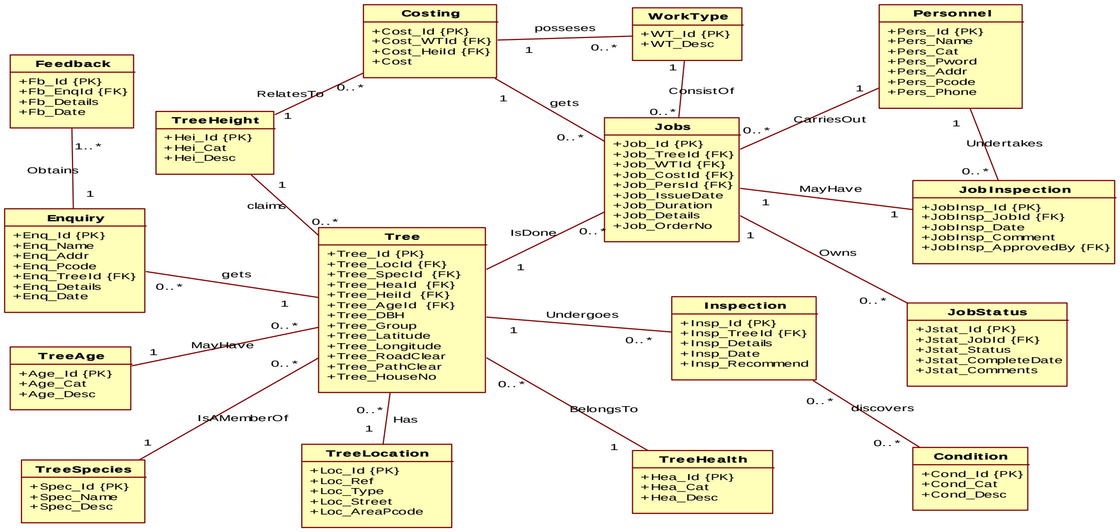

Fig. 1: Arboriculture Management Conceptual Data Model

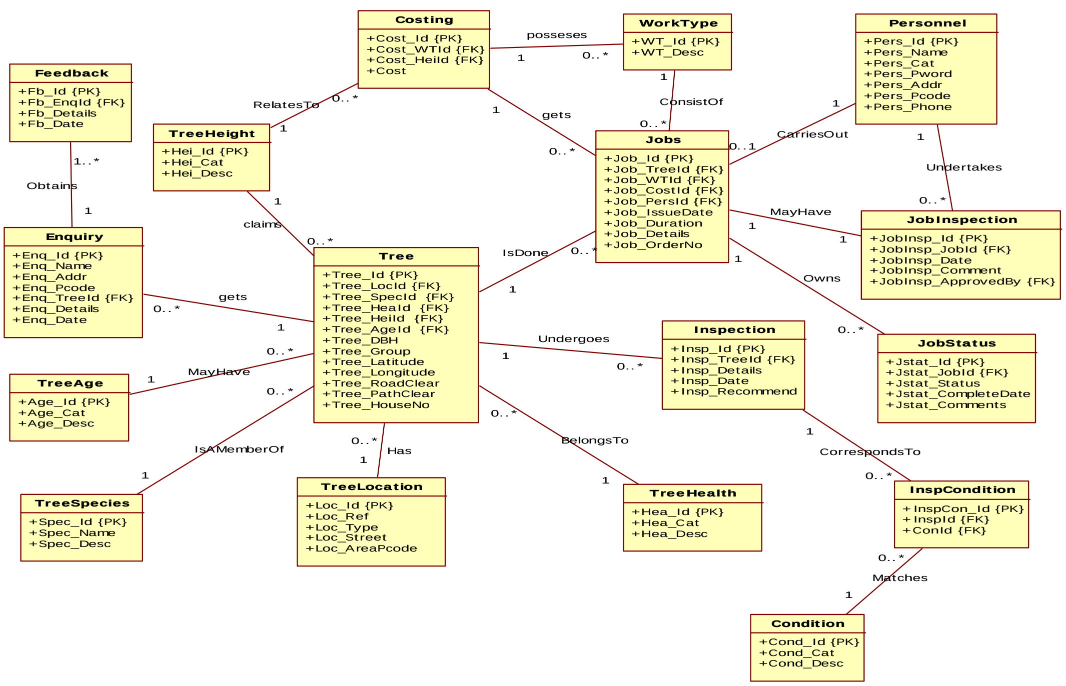

Fig. 2: Arboriculture Management Logical Data Model

Table 1: Job Table

JobId	TreeId	LocId	Date	Task	Prior	Duration	Status
1	12	5	19/04/10	Cleaning out	1	1 days	Pending
2	78	6	12/12/09	Fell and replace	2	2 days	In progress
3	321	2	25/05/09	15% Thin crown	3	6 hours	completed
4	87	4	04/04/09	Tree planting	4	3 hours	Completed
5	2	23	23/02/10	Remove obstruction	3	2 hours	Suspended
6	45	11	28/07/08	Weed	2	1 hour	In progress
7	32	13	09/08/08	Silvicultural thinning	5	3 days	Pending

A Mobile Solution for Arboriculture Management Model: Figure 3 presents SSIS model.

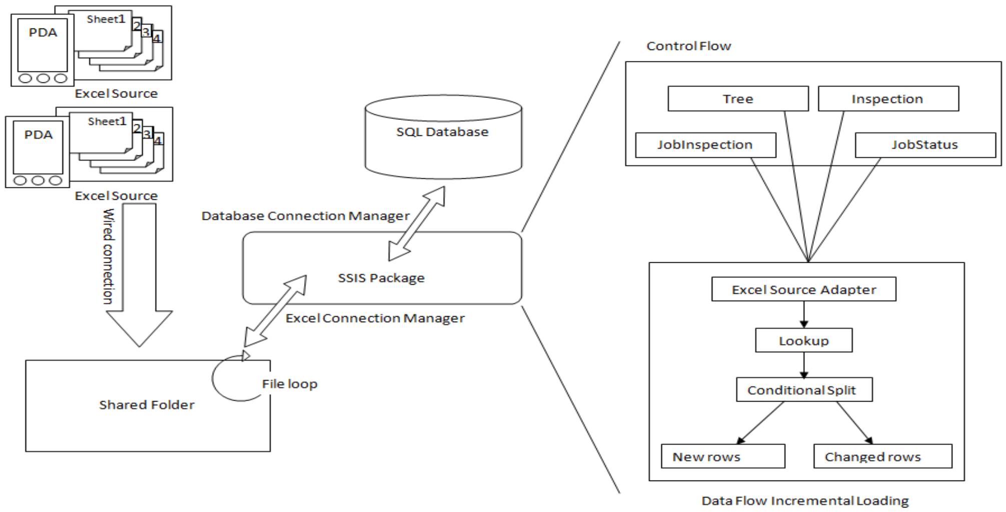

Fig. 3: A Mobile Solution for Arboriculture Management System Design

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

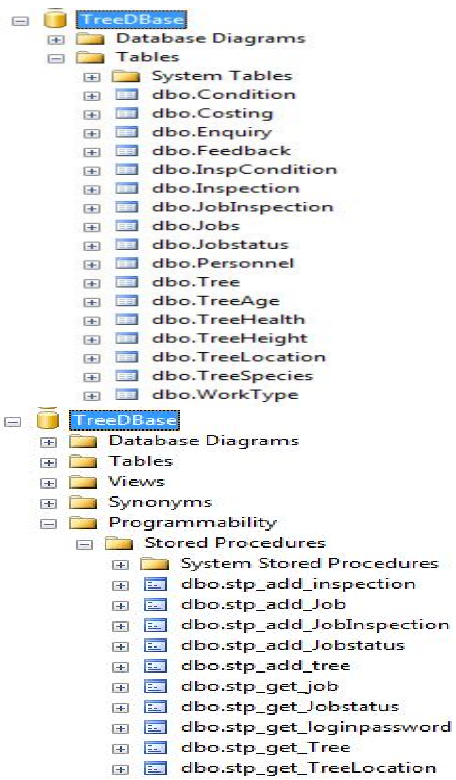

Fig. 4: Database Tables and Stored Procedures

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

Fig. 5: Original connection manager's connection string

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

Fig. 6: Updated Connection string

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 rows (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 carryout 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.

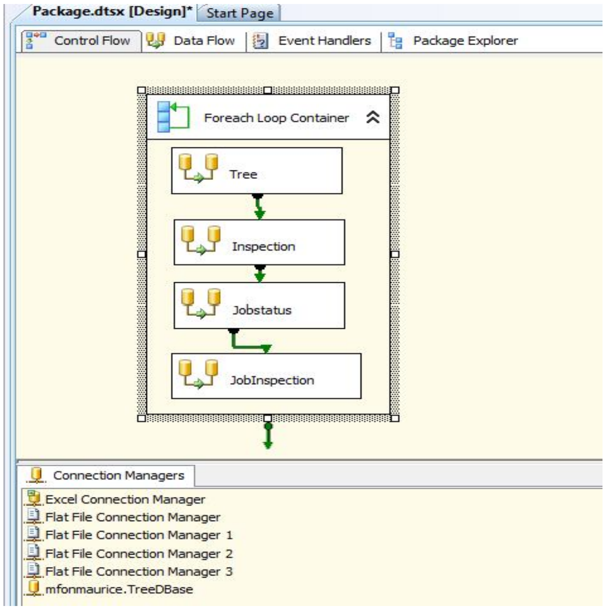

Fig. 7: Control Flow Interface

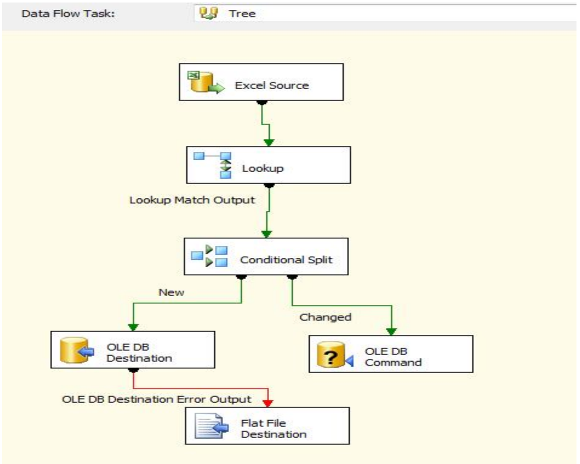

Fig. 8: Data Flow Task Showing Incremental Load Structure

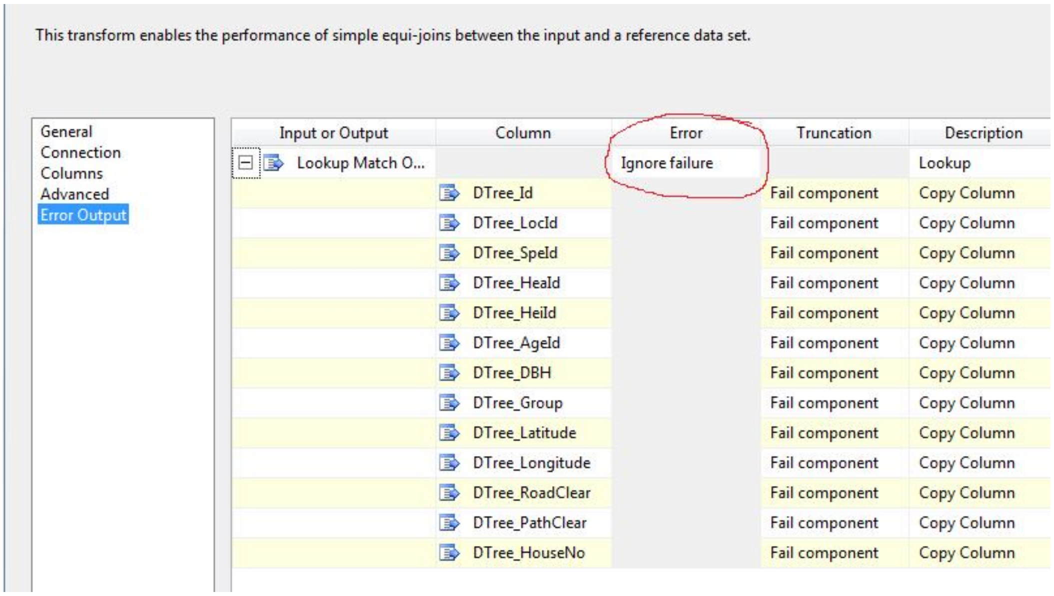

Fig. 9: Lookup Transformation Error Output

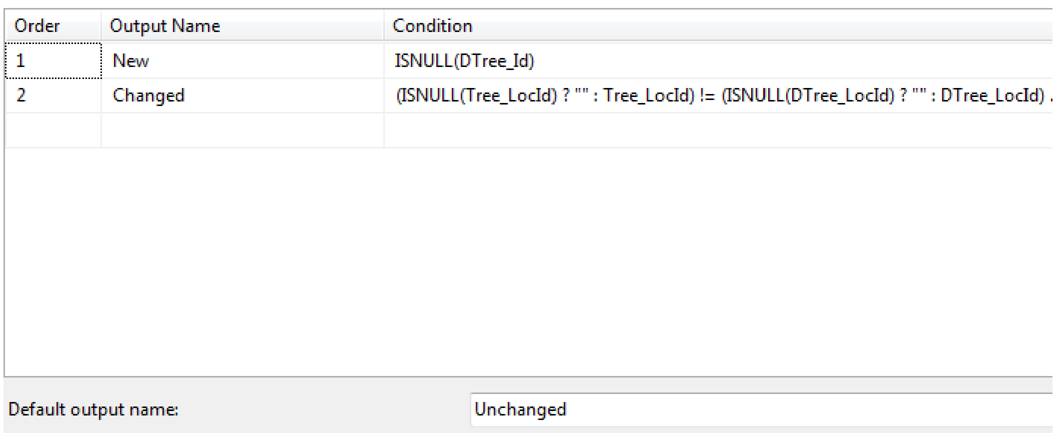

Fig. 10: Conditional Split Transform Editor

```
UPDATE [dbo].[Tree]

SET

[Tree_LocId] = ?,[Tree_SpeId] = ?,[Tree_HeaId] = ?,[Tree_HeiId] = ?,[Tree_AgeId] =
?,[Tree_DBH] = ?,[Tree_Group] = ?,[Tree_Latitude]= ?,[Tree_Longitude] = ?,
[Tree_RoadClear] = ?,[Tree_PathClear] = ?,[Tree_HouseNo] = ? WHERE [Tree_Id] = ?
```

Fig. 11: Tree table update code in an OLE DB Command Transformation.

Loc Ref	Group	Tree_No	Species	Location_T	Height_Cla	House No	Site Type	D.B.H	Rd Clearen	Path Clear	Age Class	Health Cla	Condition	Latitude	longitude
373	0	4	PRU DUL	0	E	50	G	27	2	2	SM	C	SC	54°33'53.08"N	1°14'23.25"W
373	0	5	PRU DUL	0	E	55	G	21	3	1.5	SM	C	DW	54°34'12.98"N	1°14'09.30"W
373	0	6	PRU DUL	0	E	49	G	18	3	2	SM	D	PF SD BC DW	54°34'10.45"N	1°14'06.05"W
373	0	7	PRU DUL	0	E	45	G	21	2	2	SM	C	DW SC	54°34'46.13"N	1°14'02.11"W
373	0	8	PRU DUL	0	E	41	G	19	3	2	SM	C	DW	54°34'34.63"N	1°14'26.51"W
373	0	9	PRU DUL	0	E	37	G	21	2	2	SM	C	DW	54°34'19.55"N	1°14'18.12"W
373	0	10	PRU DUL	0	E	29	G	24	2	2	SM	C		54°33'53.41"N	1°14'08.44"W
373	0	11	PRU DUL	0	E	25	G	18	3	2	SM	D	PF DW	54°33'51.99"N	1°14'27.67"W
373	0	12	PRU DUL	0	E	21	G	27	3	3	SM	B	SC	54°33'51.88"N	1°14'21.72"W
373	0	13	PRU DUL	0	E	9	G	21	2	2	SM	C	SC DW SD	54°34'44.11"N	1°14'19.02"W
374	0	1	TIL COR	0	E	5	G	9	2	2	Y	C		54°34'30.53"N	1°13'52.85"W
374	0	2	CRA X LAV	0	E	7	G	9	2	2	Y	C		54°33'58.28"N	1°14'23.25"W
374	0	3	PRU DUL	0	E	11	G	8	2	2	Y	D	CD EP	54°33'12.68"N	1°14'09.30"W
375	0	1	PRU KAN	0	E		PX	12	0	0	Y	D	BD SD ES	54°34'22.51"N	1°14'06.05"W
375	0	2	TIL EUR	0	E		PX	6	0	0	Y	C		54°34'64.23"N	1°14'02.11"W
376	0	1	PRU KAN	0	D	1	G	23	3	2	SM	C	RD BC	54°33'52.66"N	1°14'26.51"W
376	0	2	PRU SER	0	D	9	G	14	2	2	SM	C	SD SC	54°34'19.55"N	1°14'18.12"W
376	0	3	ACE PSE	0	D	17	G	18	2	2	SM	C	DW	54°33'43.71"N	1°14'08.44"W
376	0	4	CRA PUR	0	E	34	G	2	2	2	Y	E		54°33'50.69"N	1°14'27.58"W
376	0	5	CRA MON	0	E	23	G	3	2	2	Y	B		54°23'51.48"N	1°14'18.62"W
376	0	6	CRA MON	0	E	27	G	3	0	0	Y	B		54°34'20.11"N	1°14'29.09"W
376	0	7	PRU KAN	0	D	33	G	42	3	2	SM	B		54°34'30.24"N	1°13'32.15"W
376	0	8	VACANT	0	VA				0	0	Y			54°33'43.05"N	1°14'11.23"W

Fig. 12: Tree Survey Details

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

Andres, C. (2004). Tree Management Tool. M.Sc Thesis, School of Computing, University of Teesside, Middlesbrough TS1 3BA.

Connolly, T. M. and Begg, E. C. (2010) Database Systems: A practical Approach to Design, Implementation, and Management. 5th ed. London: Addison-Wesley.

Dufour-Kowalski S., Courbaud B., Dreyfus P., Meredieu C., de Coligny F., 2012. Capsis: an open software framework and community for forest growth modelling. Annals of Forest Science (2012) 69:221–233.

Dreyfus P., (2012). Joint simulation of stand dynamics and landscape evolution using a tree-level model for mixed uneven-aged forests. Annals of Forest Science (2012) 69:283–303.

de Coligny F., Dufour S., Dreyfus P., Courbaud B., Auclair D., 2010. CAPSIS: Computer-Aided Projection for Strategies In Silviculture. In Falcao. AO, Rosset. C (eds.) Proceedings of the Workshop on Decision Support Systems in Sustainable Forest Management. Lisbon.

de Coligny F., 2007. Efficient Building of Forestry Modelling Software with the Capsis Methodology. In: Fourcaud T, Zhang XP, eds. Plant Growth Modeling and Applications. Proceedings of PMA06. Los Alamitos, California: IEEE Computer Society, pp. 216-222.

de Coligny F., 2005. Capsis: Computer-Aided Projection for Strategies In Silviculture, a software platform for forestry modellers. Workshop on Information Science for Agriculture and Environment (ISAE). 3-4 june 2005, GuiZhou Normal University, GuiYang, P.R. China.

Fonseca T., Parresol B., Marques C. and de Coligny F., 2012. Models to Implement a Sustainable Forest Management - An Overview of the ModisPinaster Model. In Jorge Martin Garcia and Julio Javier Diez Casero (eds.), Sustainable Forest Management - Current Research, ISBN: 978-953-51-0621-0, pp. 321-338.

Huang, C.-Y., G.P. Asner, R.E. Martin, N.C. Barger, and J.C. Neff. (2009). Multi-scale analysis of tree cover and aboveground carbon stocks in pinyon-juniper woodlands. Ecological Applications 19:668-681.

Leskey, T. C., Pin˜ Ero, J. C. and Prokopy, R. J (2008). Odor-Baited Trap Trees: A Novel Management Tool for Plum Curculio (Coleoptera: Curculionidae). J. Econ. Entomol. 101(4): 1302-1309.

Maastikuarhitektid, K. (2012). Comparative study of tree management software. Project GreenMan (ELRI-177).

Porte´, A. and Bartelink, H. H. (2002). Modelling mixed forest growth: a review of models for forest management. Ecological Modelling 150 (2002) 141–188.

Ramakrishnan, R. and Gehrke, J. E., Database Management Systems, Third Edition, 2002. McGraw Hill.

Shamash, J. (2007) Modern Methods to Manage Trees.

Thalheim, B., (2000). Current Issues in Databases and Information Systems, East-European Conference on Advances in Databases and Information Systems. With Julius Stuller, Jaroslav Pokorný, and Yoshifumi Masunaga (eds.), Held Jointly with International Conference on Database Systems for Advanced Applications, ADBIS-DASFAA 2000, Prague, Czech Republic, September 5–8, 2000, Proceedings Springer.

Wang, Y., Bai, S., Lu, Y and Zhang, S, (2013). A GIS-Based Target Tree Management Supporting System Based on Plug-In Technology, Research Journal of Applied Sciences, Engineering and Technology 5(8): 2514-2519.

Wu, J. J. Wang, Y. Li, and B. Spong. 2012. A web-based decision support system for analyzing timber harvesting costs and productivity. North. J. of App. For. 29(3): 141-149.