High Quality Analytics with Poor Quality Data

A Case Study in Silviculture

W. Haque and J. Edwards
Department of Computer Science
University of Northern British Columbia
Prince George, Canada
{haque, edward2}@unbc.ca

Abstract—**Poor data quality has often been cited as the single most common problem hindering the deployment of Business Intelligence (BI) solutions. This problem is compounded when analytics is performed in non-conventional BI areas such as forestry and silviculture. In this paper, we describe a methodology to perform BI analytics on data that was never collected to be used for this purpose. We show that data of such low and poor quality can be transformed and loaded into the data warehouse which is then used for high quality reporting.**

Keywords-Analytics; Business Intelligence; Data Quality; Silviculture

I. Introduction

The use of online analytical processing (OLAP), statistical analysis, forecasting and data mining now extends beyond conventional enterprises which use such information to make strategic business decisions. The practice of query and reporting using visualization tools can be found in many non-traditional applications, such as forestry and silviculture. The key difference, however, is that the latter uses this information primarily for research and analysis as opposed to financial gains or competitive advantage [1]. Further, in non-traditional analytic applications, the data is not gathered with specific OLAP queries in mind.

The data collected over long periods of time without specific goal in mind is usually a low quality or noisy data which consists of incomplete, incorrect, inconsistent and duplicate values, to name just a few issues. Data in such a form cannot be readily used for analysis as the results would not be reliable and accurate. Hence, it is desirable that such data must be cleaned and transformed into high quality data before being loaded into a data warehouse (DW) [2] which must act like a single version of truth for the organization [3]. The data in DW is focused, integrated, and non-volatile. This pool of structured data is then used for query analysis and to generate reports which support decision making. The process of extracting the data, converting it into a high level data, and finally loading into DW is known as the ETL (Extract Transform Load) process. ETL is the most time-consuming process in the entire BI deployment. For data that is collected without clear specifications from multiple sources, this step becomes even more difficult [4].

The data used in this case study is silvicultural data representing over 800 installations managed by the Northern Interior Vegetation Management Association (NIVMA), a forest industry co-op initiated by British Columbia and Alberta forest product companies. Silvicultural data includes information regarding establishment, growth, health, composition, and overall quality of trees and forests to meet the needs of the forest industry. The NIVMA data was originally recorded for the purpose of generating models which could predict stand growth under specific conditions. As stated earlier, this data suffered from the typical quality issues and was not ready for uploading to the data warehouse in its current form.

In this paper, after presenting some work related to data quality, we discuss the challenges posed by NIVMA data. We then describe how these challenges were addressed and an acceptable OLAP cube was designed. The ETL process specific to this database is then presented in detail. We conclude by providing some representative analytical reports generated from the cube.

II. Related Work

The undesirable quality of raw data is an issue present in any data collection regardless of whether single or multiple data sources are used. Due to the practical importance, data cleansing and related topics are receiving more and more attention in the research community. A general classification of quality problems which can be addressed by data cleansing is provided in [5]. The authors identify ways in which inconsistencies may occur based on either the source (single or multiple) or the location (schema level or instance level), or both. The general steps required in cleaning data that is meant to be used for analysis are then defined. These steps include initial data analysis, the definition of transformation workflows, verification, transformation, and in some cases the backflow of improved data into the original source.

A general process for integrating multiple databases is outlined in [6]. Although we are not fully integrating data from multiple databases, the data integrity issues presented by the NIVMA database are largely similar to those which appear in scenarios where databases are being combined. The main

DOI: 10.5176_2010-2283_1.4.100

steps outlined in the merging of independent databases include pre-integration, correspondence identification, and the integration stage itself. The integration steps outlined in the paper are particularly relevant to our work with NIVMA.

A methodology for cleansing heterogeneous databases is given in [7]. The scenarios described in this paper generally involve cases such as an organization which needs to add operational data from a new source to an existing data store, creating heterogeneous entries. This provides a good analogue for the data in NIVMA, which has been collected by various agencies with varying standards for some entries.

With respect to forestry data, work is being done to generate simulations of future stand growth with software such as TASS III [8] or TIPSY [9], although a major weakness in these models is the lack of appropriate data for some forms of analysis [10]. These models provide a fairly accurate prediction of stand growth and yield based on conditions presented; however, they also work using parameters which are beyond the scope of the data in the NIVMA data warehouse, such as volume merchantability limits and specific plot dimensions.

III. Data Challenges

Due to the fact that NIVMA data is not standard business data, the ETL process is presented with several unique challenges. Perhaps the most striking difference with this data is the way in which it was collected. Over the years, several forestry agencies had undertaken the task of collecting and aggregating this data, which has created integrity issues within many of the database tables due to inconsistent procedures and specifications. A second problem which becomes immediately apparent is the consistency (or lack thereof) with which measurements are taken. The number of measurements per tree and the ordering of those measurements vary wildly, which can become an issue if we attempt to analyze trees at a high level. There is also no standard for the starting age of a tree, though the analysis requires knowledge of how long a tree has been growing. The issues occurring as a result of data being recorded heterogeneously by these multiple agencies are very similar to those which would be experienced by an organization integrating more standard data from multiple sources as outlined in [6]. For reference, we provide solutions used for some of these problems before describing the ETL process in detail.

As an example of the lack of standard units of measurement, we note that the majority of trees have an initial height measurement that is less than 100 while some trees have an initial height that is over 1000. Generally, the larger trees exhibit variance by a factor of ten, which has led us to the assumption that the larger trees were measured using a different unit. Such a significant variance creates obvious problems for analysis, for example if examining the average height of all trees within a certain species is valuable, this skewed average could lead to misinformed decisions at the end user level. In order to compensate for differences in unit of measurement, growth rate is examined using the measurements of growth over previous years at a tree level. This percentage is calculated for all trees within a stand, and is then aggregated at higher levels to give an average for larger groups of trees. A second issue with the height measurements is the way in which a dead tree is represented. In most cases a value of zero indicates a dead tree, but in a few cases a dead tree has been indicated with a value of 9999; this once again can potentially skew any aggregation. We solve the first issue by making the assumption that the different heights occur in either cm or mm, and we scale the size of a tree accordingly. Approximately 0.5% of trees in the database appear disproportionately large, and thus the overall effect of this change should only be strongly apparent in specific tree groupings (such as species) which are dominated by these very large trees. The dead tree issue is resolved in a similar manner, by simply setting the trees with a 9999 height to a value of zero.

In the NIVMA database, individual trees are measured on a semi-yearly basis; however, the year of a first measurement in each block is not consistent amongst all blocks. A further complication occurs in that there is a large variance between the sizes of first height measurements for trees which seems to indicate that they may not all be measured after the same period of growth. At a specific block level this does not create problems as the vast majority of blocks have a consistent measurement pattern. At a high level however, we want to be able to aggregate all of the growth data for a species and from that be able to make assumptions about the typical growth patterns for this species. We then are able to use these high level growth trends to identify specific blocks that are performing poorly or exceptionally, and identify the individual factors that may have led to these growth patterns (Fig. 6). Occasionally there are also instances where stands are measured inconsistently in terms of the time period over which measurement is taken. For example, there exist stands for which two sets of measurements have been taken, each representing a different group of trees within the stand. When such an instance occurs, each set of data is usually recorded over contiguous but mutually exclusive time periods. For example, one set of trees may be measured beginning in 2000 and ending in 2004, and another set within the stand will be measured beginning in 2004 from where the measurements on the other group ended. When analyzing data at a stand level, this creates an issue as a given year of measurements may contain both very large and very small trees should one of these measurement groups have a crossover in time period.

Beyond problems presented by individual trees, there are issues which occur in trying to relate stands of trees to larger logical groupings, the largest of which being a block. Specifically, these issues occur when attempting to make sense of metrics which are tied to large areas as opposed to the trees therein, such as density of trees or soil conditions. Each Block is comprised of a set of Quads which are themselves composed of a set of specific Grids. These Quads and Grids are used to identify the location of each recorded measurement within a Block, and therefore are one of the key ways to logically connect the low-level measurements. There are two key issues which arise within the structure of the NIVMA database tables when we try to associate repeatedly measured trees with density data. The first is that, in a few cases, the Grids and Quads are not related to one another properly for either the density or tree data due to some keys being entered incorrectly. The second, more serious, problem is that not every group of repeatedly measured trees has a recorded density measurement, and likewise there exist density measurements which cannot be tied to a specific set of trees. As a result, when we try to create a regular relationship between the required tables based on some form of Grid/Quad key there are instances where no foreign key exists. We solved this consistency issue by inserting several null records into the Grids and Quads tables where necessary in order to ensure that a foreign key relationship could be established. This has resulted in adding a relatively minimal 2150 null records to a table consisting of 36,252 records.

IV. Cube Design

An OLAP data cube is essentially a representation of the underlying data warehouse in a manner that allows efficient analysis of data. The cube allows queries on aggregated data with improved performance over a data warehouse or data mart alone [11]. Thus, before building a data warehouse from the NIVMA data, it was important to determine the final cube design with reference to the perceived analysis. The most useful feature of an OLAP cube is its ability to "slice" data along the dimensions identified for analysis by the end user. The dimensions which we have determined to be most valuable for our analysis include: Agencies responsible for individual trees or stands, time at which measurements are taken, soil features, tree species, location, and ecological zones. The Agencies dimension includes codes representing each agency, full agency names, information about which plots belong to which agency, and other operational data. Time is relatively self-explanatory as a dimension and contains days, months, and years. The Soil dimension contains very specific information about plot soil conditions, such as percentage of gravel present or the depth at which any restricting layers may occur. Species is a comparatively simple dimension, consisting of the species code, full species description, and other species related facts such as whether the species is coniferous or deciduous, or whether the species is a crop tree. Location contains basic information about which block, quadrant, and grid a plot or tree belongs to with respect to the British Columbia geographic system of mapping [12]. Ecological zones refer to areas of BC with uniquely defined ecosystems, which allows the dimension to be defined in a hierarchy containing zones, subzones, and ecozones. The data which we are slicing using these dimensions is split into two logical groups: one which pertains to individual repeatedly measured trees, and the other which represents entire stands of trees. These are organized into fact tables joined to the dimensions by one foreign key for each dimension table, and then aggregated to create a composite key for the two fact tables.

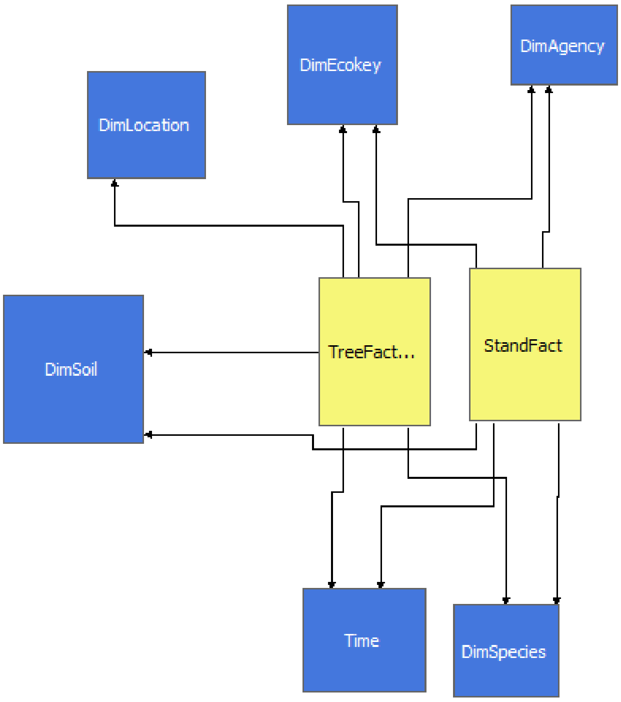


Figure 1. NIVMA cube design

Once the dimensions and fact tables have been determined, the only remaining decision is the schema to be used for the data warehouse structure. The two schemas to choose from are the star and snowflake schema [11]. Both schemas link the fact tables to dimensions in the way described above except for one key difference. Generally, a dimension can often be split into multiple levels forming a hierarchy. In the snowflake schema, the dimension tables are split into separate tables for each level of the dimension hierarchy in a way similar to normalization in a relational database. Consider, for example, a time dimension that would contain individual tables for years, months, and days. In the star schema, dimension hierarchies are not separated into multiple tables and remain de-normalized, creating faster query execution due to fewer table joins being necessary. The purpose of dividing the dimension tables is primarily for maintenance purposes. The new data for this application (NIVMA) is expected to arrive periodically and not nearly at the rate that is common in a standard business. Thus, future maintenance is not a significant issue for this project and we chose to use the star schema.

Beyond the structure of the cube there is also the option to add calculated measures for the use of reporting in the final stages. In this scenario there are actually several metrics which are not immediately present in the data but can be extracted using some calculations. For example, due to the differences in units of measurement as mentioned in the previous section our growth rates for stands are calculated with an MDX (Multidimensional Expressions) query to avoid integrity issues which would arise from simply aggregating the growth fields presented in the database. Other metrics which require calculation through the use of more complex expressions include the brush competition index and the survival rating of a tree. The Comeau brush competition index compares the height and coverage of surrounding brush against the average height of trees. This presents similar issues to growth rates in varying measurement units as seen before and adds the issue of unifying data from several columns which contain information relating to the types of brush in a specific area. Survival rate is calculated using the 'vigour' rating of a tree over the period of time for which it was measured. A tree may have a vigour of one, two, or three; each of those numbers representing death, some intermediate status such as damage or disease, or strong health respectively. For our analysis we look at the final vigour of a tree during its associated measurement period and compare that against the number of unique trees in the stand, providing a survivability rating which proves useful at a single tree or higher level analysis.

One area of analysis which we have chosen to leave out of the project at this point is some form of damage/treatment dimension. Within the NIVMA database there is information related to any recorded damage or currently applied treatments which occur in a stand of trees. This could be anything from fire damage to pine beetle infestation. Of course, this information could be leveraged in order to supplement the survivability rate as described earlier, or even to forecast the quality of wood produced. However, the level of silvicultural knowledge and technical work required in normalizing this data and extracting useful information while still identifying inconsistencies and poorly collected records is beyond the scope of this research.

V. The ETL Process

We have used SQL Server Integration Services (SSIS) to perform the ETL process. SSIS is a component of Microsoft's SQL Server database software, which provides a convenient platform for data integration as well as data warehousing tools that assist in the ETL process. The overall SSIS process for creating the NIVMA Data Warehouse is shown in Fig. 2, with the order of processing beginning with truncation of all tables to ensure that data is not added in duplicate if the process is run more than once, and order following that being illustrated by the upward pointing arrows. Any disconnected transformations proceed in parallel with other transformations in the same step.

The left process container in Fig. 2, labeled "Populate Dimensions" represents population of each of the dimensions in parallel, and although the order in which they are populated doesn't affect the process they must all be populated before the fact table can be filled. Each of these dimensions is described below.

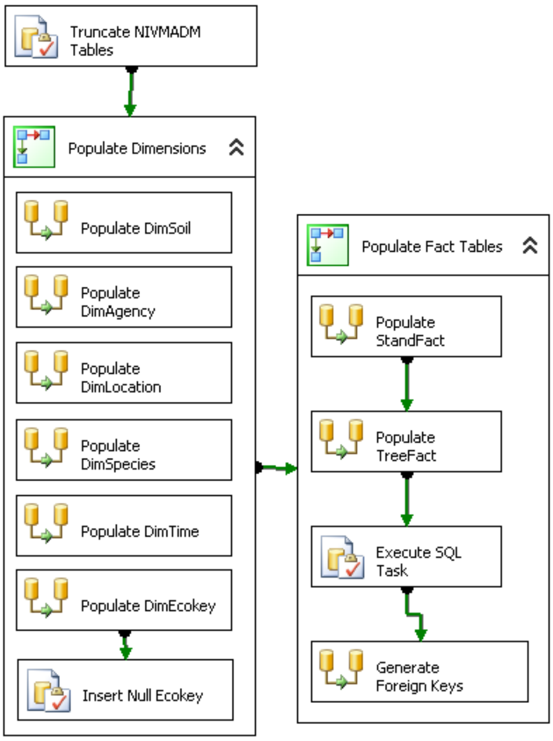


Figure 2. The overall SSIS process

A. *Populate DimEcokey*

The DimEcokey table of the data warehouse draws primarily from the NIVMA table which lists the zone, subzone, and variant of each plot. From this table, the first step is to replace all NULLs which appear in the variant column with a blank space, as the derived column transformations provided in SSIS will otherwise return a null string when NULLs are concatenated with other strings. Next we generate the ecokey of the plot which is simply the concatenation of zone, subzone, and variant. This is also why it is important that there be no NULL values in variant, as this could potentially result in a NULL ecokey as opposed to an ecokey with a blank variant. Next the table is sorted by ecokey to allow for a merge join with a table from a BC forest service website [13] which lists zones, subzones, and variants with useful descriptions. This BC forest service table is saved as a comma separated value file and we load it as a flat file source. Similar processing is performed on the data from this table so that we have an ecokey which is similarly sorted to participate in the merge join.

Following this, we are ready to fill the DimEcokey table. The general process is shown in Fig. 3. It should be noted here that once the table is filled, an additional SQL command is required as an external component to insert a row with a blank ecokey in rows of tree and stand data for which no ecozone is specified; otherwise, if an attempt is made to look up data from the ecokey dimension for a row in which no ecokey is available an error could occur. Other dimensions generally look very similar in terms of the ETL process due to repeated use of derived columns and sort transforms followed by merge joins.

B. *Populate DimTime*

The DimTime table of our data warehouse draws from the repeatedly measured tree header table of the NIVMA database. This table contains data for each planted stand, the most important part of which being the year, month, and day each one was planted. The first step preformed is generating the date column which holds the primary key date. The function used simply concatenates the year, month, and day to give a date in the DD/MM/YYYY format. Following this, data is sorted by date and then all values are converted from Unicode strings to non-Unicode strings to match the data types in our data warehouse.

The last component before loading into the data warehouse is a script component which maps the column containing the numeric month (1-12) to a string ("January" through "December") in the month description column. The DimTime Table can now be loaded with data.

C. *Populate DimSpecies*

Two sources are used in this dimension; the Alberta species and BC species tables which contain the species measured in these two provinces, respectively. Little processing of data is required here as the structures of both are the same as the dimension table structure, so the tables are merged together to create a table of their combined rows. These two tables also contain other useful species pertinent information such as a species status as harvest or non-harvest, and whether a tree is deciduous or coniferous. Following this, the data is sorted to remove duplicate species and loaded into the DimSpecies table.

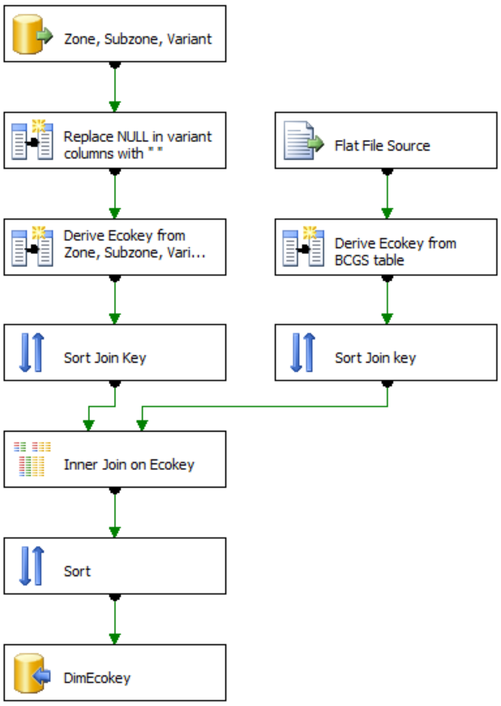

Zone, Subzone, Variant
Replace NULL in variant columns with " "
Flat File Source
Derive Ecokey from Zone, Subzone, Vari...
Derive Ecokey from BCGS table
Sort Join Key
Sort Join key
Inner Join on Ecokey
Sort
DimEcokey

Figure 3. Populating DimEcokey

D. *Populate DimLocation*

The header table for repeatedly measured trees contains the basic information for each tree, which includes the quadrant and grid number which applies to individual trees. The quadrant description table contains descriptive information such as slope and angle at each quad. This allows us to select trees in areas ranging in size from grids and quadrants, to individually measured trees.

E. *Populate DimAgency*

This step is one of the most simple of the processes in this SSIS package. An inner join is performed on the administrative header and agency tables using the "Agency" string as the join key. The administrative header table provides the agency information for each individual plot, and the agency table provides details for each agency; combined, this gives us all we need for our dimension and fill the DimAgency table.

F. *Populate DimSoil*

DimSoil is by far the most complicated dimension in our data warehouse. As such, we have split the process into three sections which can be dealt with independently and then joined. To begin this process, we extract soil data from the soil header which contains identifying information for pits, stands etc. This data is used to create a composite key, soilkey, which uniquely identifies a soil measurement. The soil restricting layer depth table contains values for soil restrictions and their respective depths. This table is joined with the soil restriction type table which contains descriptive values for each soil restriction type, such as solid rock or gravel. Now both of these tables can be joined after a soilkey column is created from the restriction type data.

The second part of this process uses data from the soil humus table. Humus is the top level or horizon of soil. Soilkey is derived from this table as before, and the data is then merged with data from the soil horizon and soil modifier tables. The soil horizon table contains descriptive data for the types of soil horizon which appear in the soil data. With each horizon layer there are also modifiers which describe unique attributes within one type of horizon, descriptions of which are provided in the soil modifier table. After these tables are joined on horizon and modifier types, respectively, we can combine this part of the process with the others using a merge join on soilkey.

For the last part of the process, we note that the data source contains data describing all horizon layers below the humus layer. This data source is joined with the root table which describes the root presence in the soil as denoted in the soil horizon table. Likewise, the texture table contains descriptions for the texture values denoted in the soil horizon table. Once these descriptions have been added, we can merge join this data with the previously joined data in order to add the horizon layer descriptions.

Finally, all three parts are merged and loaded into DimSoil.

G. Populate StandFact

Now that all of the dimensions have been filled, we are ready to begin filling our fact tables. The stand data requires a somewhat simpler ETL process, as stand level data is held in only three main tables: the plot header table, tree measurements table, and damaging agents table. We join these tables using a unique identifier composed of the OBLIGREF number, the year in which the measurements were taken, the harvest reference number for the stand, and the stand identifier. We create this key in a new column with a derived column transformation in SSIS. This column then contains a string representing the unique identifier. A full outer join on the three data sources using a merge join transformation is then performed. The StandFact table can now be filled.

H. Populate TreeFact

The final step in the overall process is filling the fact table, TreeFact. Once again, we begin with several tables that need to be merged. The header table contains basic administrative information for the repeatedly measured trees. There are six other tables which contain reference numbers for the trees; the BC stocking standards number, Alberta stocking standards number, harvest reference number, artificial regeneration number, stand tending reference number, and the site preparation reference numbers. Up to this point the process is relatively simple, requiring only basic joins. Next, we must add the repeated measurements.

Given the number of records being joined, we actually store our merged data in a temporary staging table, and then perform the outer join on the repeated measurements using an SQL query as opposed to an SSIS sort transformation for improved efficiency. Once this step is complete we enter another data transformation task where we first merge join the data in our staging table with annual increment measurements and then begin to add our composite keys using derived column transformations as in the stand level fact table. The second portion of this fact table population is shown in Fig. 4.

VI. EXPERIMENTAL RESULTS

Once the data is in data warehouse structure, it can be loaded into the OLAP cube for analysis. The analysis is performed using Microsoft's SQL Server Analysis Services (SSAS) and reported via SQL Server Reporting Services (SSRS). There are several ways that data can be viewed to create useful reports for operational analysis, across any combination of our six dimensions. One such report that fully utilizes the abilities provided by our OLAP cube is shown in Fig. 5.

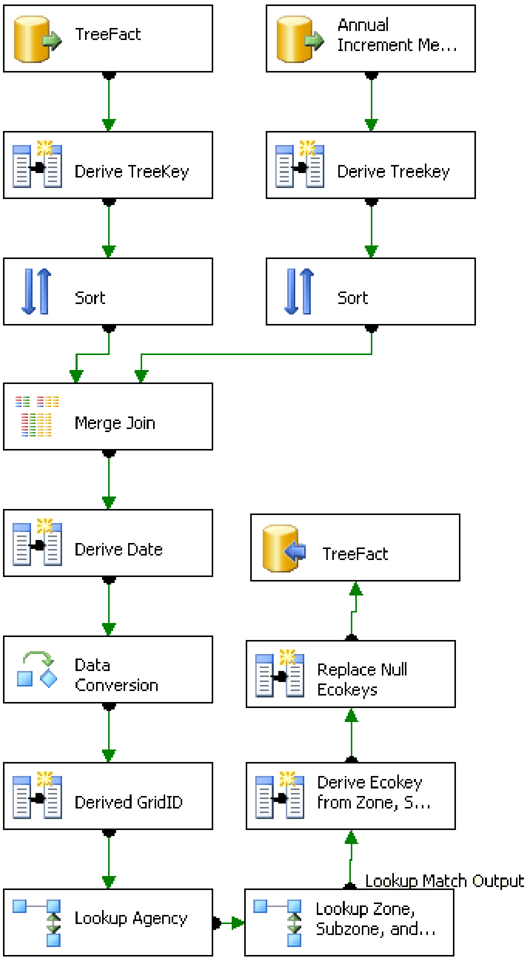


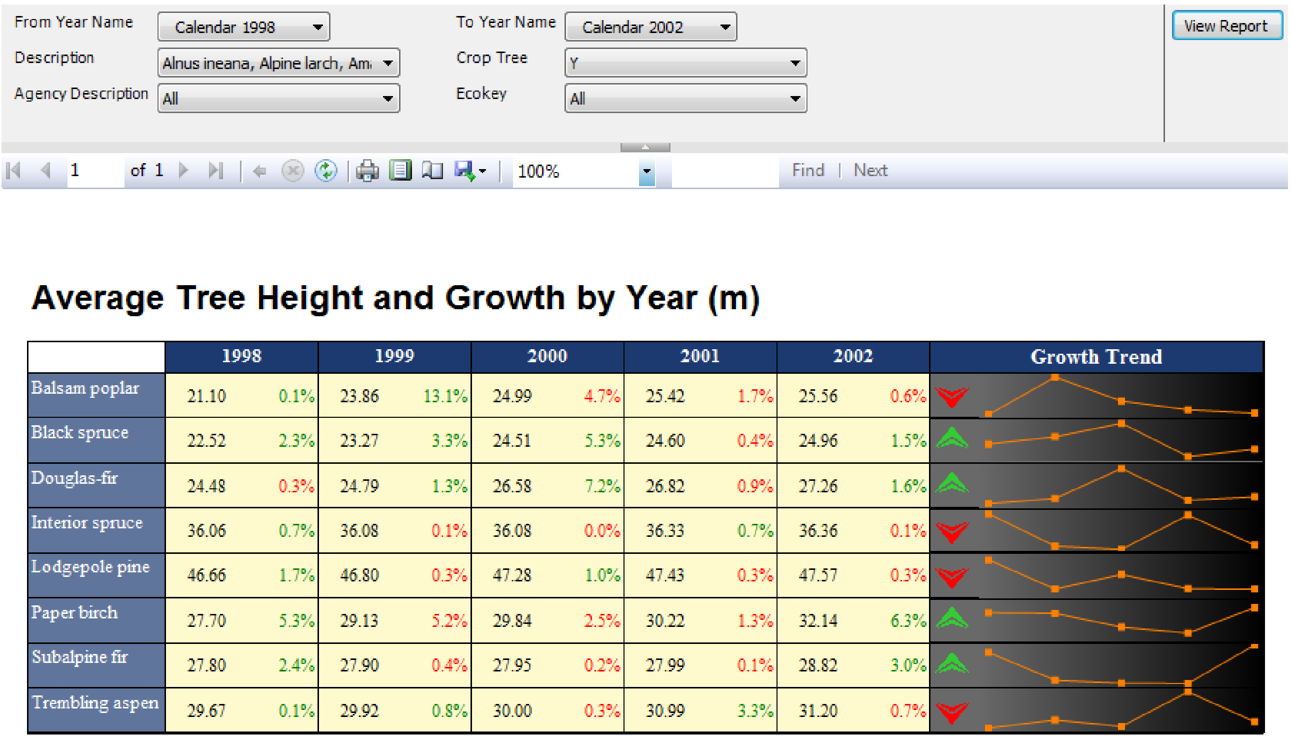
Figure 4. Population of TreeFact with Annual Increment Measurements

The most important questions when choosing where and how to allocate resources in planting often involve determining where the highest output will be found in terms of lumber volume. Given this knowledge combined with the historical data at hand, the chart in Fig. 5 can be used to gain some valuable insight. In each cell the average height across all individually measured trees of the corresponding species is shown, as well as the increase or decrease in over the previous year's average. If data is not available for a particular species in a given year, the cell simply grays out and is not considered in the percentage change calculation for the next year. Following this, there is a trend column which includes an indicator showing the growth trend as positive or negative over the last year, as well as a Sparkline which gives a visual idea of the growth pattern over the years.

The most important part of this chart is the parameterization of its dimensions. In the uppermost portion of Fig. 5 there are drop-down menus visible on the report, allowing the user to select the range of years being examined, species of tree, specific sets of ecozones, silvicultural agencies, and crop or non-crop trees. This enables the chart to dynamically give the user data for more specific scenarios, provided these scenarios have occurred in the past and sufficient data is available for them. This could be used for either identifying beneficial combinations of species and ecozone, or determining which combinations have done poorly in the past [14].

The chart shown in Fig. 6 illustrates the use of data for repeatedly measured trees. This chart is also parameterized by species, and provides insight into the features related to specific outlying trees of interest. This allows a silviculturist to select a species for which growth is particularly relevant, and view some features which may have affected the growth seen for future reference. Particularly, the data shown in this chart is related to the specific location of the outlying tree and its ecological conditions.

In the example shown in Fig. 6, we see one of the trees with the largest growth over the previous year is in the "ICHvk2" ecozone, this may lead us to the conclusion that Douglas Fir grows particularly well here. What this ecozone actually represents is the zone "Interior Cedar – Hemlock" (ICH), subzone "Very Wet, Cool" (vk), with the "Slim" variant (2). However, when we examine the trees with the

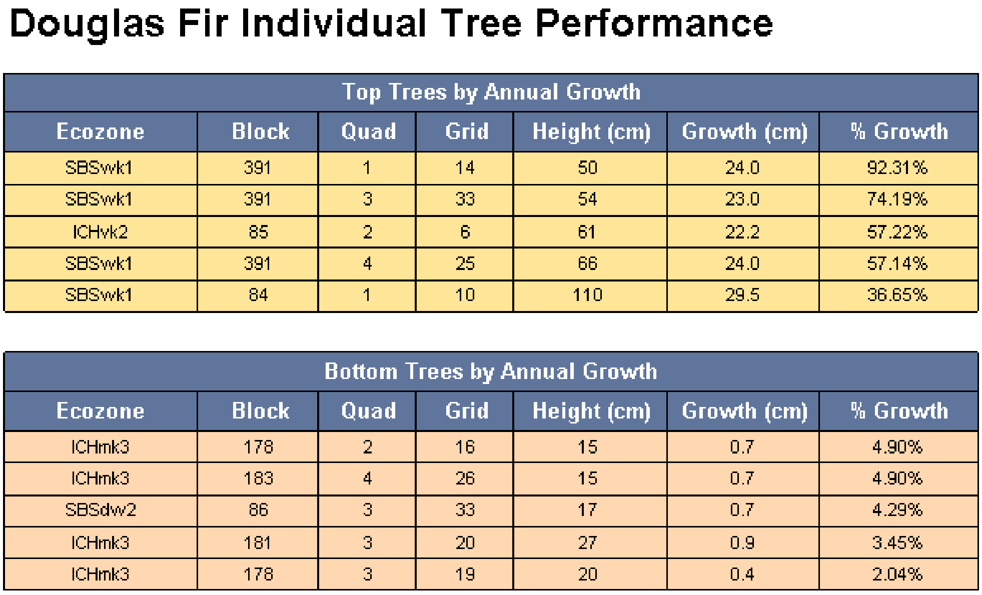


From Year Name: Calendar 1998
To Year Name: Calendar 2002
Description: Alnus ineana, Alpine larch, Am
Crop Tree: Y
Agency Description: All
Ecokey: All

Average Tree Height and Growth by Year (m)

	1998		1999		2000		2001		2002		Growth Trend
Balsam poplar	21.10	0.1%	23.86	13.1%	24.99	4.7%	25.42	1.7%	25.56	0.6%	▼
Black spruce	22.52	2.3%	23.27	3.3%	24.51	5.3%	24.60	0.4%	24.96	1.5%	▲
Douglas-fir	24.48	0.3%	24.79	1.3%	26.58	7.2%	26.82	0.9%	27.26	1.6%	▲
Interior spruce	36.06	0.7%	36.08	0.1%	36.08	0.0%	36.33	0.7%	36.36	0.1%	▼
Lodgepole pine	46.66	1.7%	46.80	0.3%	47.28	1.0%	47.43	0.3%	47.57	0.3%	▼
Paper birch	27.70	5.3%	29.13	5.2%	29.84	2.5%	30.22	1.3%	32.14	6.3%	▲
Subalpine fir	27.80	2.4%	27.90	0.4%	27.95	0.2%	27.99	0.1%	28.82	3.0%	▲
Trembling aspen	29.67	0.1%	29.92	0.8%	30.00	0.3%	30.99	3.3%	31.20	0.7%	▼

Figure 5. Average Tree Height and Growth



Douglas Fir Individual Tree Performance

Top Trees by Annual Growth						
Ecozone	Block	Quad	Grid	Height (cm)	Growth (cm)	% Growth
SBSwk1	391	1	14	50	24.0	92.31%
SBSwk1	391	3	33	54	23.0	74.19%
ICHvk2	85	2	6	61	22.2	57.22%
SBSwk1	391	4	25	66	24.0	57.14%
SBSwk1	84	1	10	110	29.5	36.65%

Bottom Trees by Annual Growth						
Ecozone	Block	Quad	Grid	Height (cm)	Growth (cm)	% Growth
ICHmk3	178	2	16	15	0.7	4.90%
ICHmk3	183	4	26	15	0.7	4.90%
SBSdw2	86	3	33	17	0.7	4.29%
ICHmk3	181	3	20	27	0.9	3.45%
ICHmk3	178	3	19	20	0.4	2.04%

Figure 6. Douglas Fir Individual Tree Performance

least growth we can see that the come from the same zone. This implies that perhaps the zone is not the cause for growth, but that Douglas Fir will perform better in very wet conditions, than in moist conditions (indicated by mk in ICHmk3). We could also say that the “Sub-boreal Spruce” (SBS) zone is not a boon or handicap to growth, as trees from this zone can be seen in both tables, however trees with high growth were in a “Wet Cool” (wk) as opposed to the tree with third lowest growth, which was in a “Dry Warm” (dw) subzone, opposite subzones having produced opposite results. In summary these observations are an indicator that Douglas Fir likely performs best in conditions with high amounts of moisture, as well as in cooler areas. At this point, we could also examine all trees in the corresponding blocks, quadrants, and grids in more detail if the results we are seeing should need any further analysis. For example our parameters could be adjusted to show general trends for the “IHCvk2” zone, and development of other trees in the same area. Other possibilities for reporting with NIVMA include histograms for metrics such as average height or growth by species and stand, gauges for key performance indicators such as survivability, and distribution charts for features such as treatments which may be present in some but not all trees within a stand. The reporting can also be tailored to a specific realm of silviculture, such as an analysis of projected wood volume being produced in a specific large area consisting of a group of stands, or a survey of a geographically distributed ecozone-species combinations of specific interest with a focus on the results of treatments with respect to annual growth.

VII. Conclusions

In this paper, we have described the issues facing development of a business intelligence solution when using unconventional data. These difficulties are compounded by poor data quality issues which are commonly present in standard data sources [2]. We have suggested solutions to some of the common problems and inconsistencies that are present in such scenarios. We have also demonstrated a methodology to construct an effective data warehouse with a representative, considerably unclean, data source found in forestry applications. This data warehouse was used as the basis for building an OLAP cube, which then generated useful analysis on the data for operational insight which can be used to reduce costs and improve yields by better site preparation and using better choices of stock type. Similarly, the effects of herbicide treatments and identification of productive and resilient forests can be made.

So far unconventional uses of business intelligence such as silvicultural analysis are uncommon; however, with the increasing popularity of analytics in all arenas and evidence that it can be performed on less than perfect data, the future looks promising.

Acknowledgment

We would like to thank Derek Belsham of Westcreek Ltd for facilitating access to the NIVMA database and providing a primer in understanding terminology specific to silviculture.

References

[1] Robert M. Frank, William B. Leak Paul E. Sendak, "Long Term Data from Silvicultural Studies: Interpreting and Assessing Old Records for Economic Insight," *Long Term Silvicultural and Ecological Studies: Results for Science and Management*, pp. 203-207, 2006.

[2] Hongjun Lu, Tok Wang Ling, Yee Teng Ko Mong Li Lee, "Cleansing Data for Mining and Warehousing," The National University of Singapore, Singapore, Technical Report TRA6/99, 1999.

[3] Efraim Turban, Ramesh Sharda, and Dursun Delen, *Business Intelligence: A Managerial Approach*, 2nd ed.: Prentice Hall, 2010.

[4] Salvatore J. Stolfo Mauricio A. Hernandez, "Real-World Data is Dirty: Data Cleansing and the Merge/Purge Problem," *Data Mining and Knowledge Discovery*, pp. 9-37, December 1997.

[5] Hong Hai Do Erhard Rahm, "Data Cleaning: Problems and Current Approaches," *Bulletin of the Technical Committee on Data Engineering*, vol. 23, no. 4, pp. 3-13, December 2000.

[6] Stefano Spaccapietra Christine Parent, "Issues and Approaches of Database Integration," *Comm. ACM*, vol. 5, no. 41, pp. 166-178, 1998.

[7] R. J. Miller, B. Niswonger, M. Tork Roth, P. M. Schwarz, E. L. Wimmers L. M. Haas, "Transforming Heterogeneous Data with Database Middleware: Beyond Integration," *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, p. 6, 1997.

[8] Jim W. Goudie, Steve Stearns-Smith C. Mario Di Lucca, "TASS III: A new generation growth and yield prediction model for complex stands in British Columbia," in *World Forestry Congress*, Buenos Aires, 2009.

[9] C. Mario Di Lucca, "TASS/SYLVER/TIPSY: Systems for predicting the impact of silvicultural practices on yield, lumber value, economic return and other benefits," in *Stand Density Management Conference: Using the Planning Tools*, Victoria, 1999, pp. 7-16.

[10] Shongming Huang W.R. Dempster, "Enhanced Fibre Production and Management of Lodgepole Pine," Joint CIF/SAF AGM, 2004.

[11] Brian Larson, *Delivering Business Intelligence With Microsoft SQL Server 2008*, Robert M. Brucker, Ed. United States of America: McGraw-Hill, 2009.

[12] GeoBC Crown Registry and Geographic Base Branch. BCGS Map Numbering System. [Online]. http://archive.ilmb.gov.bc.ca/crgb/about/bcgs/

[13] British Columbia Forest Service. Ministry of Forests and

] Range - Research Branch Web site. [Online]. http://www.for.gov.bc.ca/hre/becweb/resources/codes-standards/standards-becdb.html

[14] Fabio Casati, Umesh Dayal, Ming-Chien Shan Daniela Grigori, "Improving Business Process Quality through Exception Understanding, Prediction, and Prevention," in *Proceedings of the 27th VLDB Conference*, Roma, Italy, 2001, p. 10.

Dr. Waqar Haque is Professor in the Department of Computer Science and School of Business at the University of Northern British Columbia, Canada. His core research encompasses areas in high performance computing including real-time database systems, parallel and distributed computing, and VLDBs (very large database systems). In addition, he is involved with community projects and award winning industrial collaborative research involving business intelligence and advanced analytics. Dr Haque's research has been supported by Natural Science and Engineering Research Council (NSERC) and the industrial partners.

Jake Edwards is a fourth-year undergraduate student pursuing a degree in Computer Science at the University of Northern British Columbia, Canada. He has been an active member of the Business Intelligence Research Group, serves on the University Senate and a number of other Committees. Jake plans to pursue his Masters with a focus on advanced analytics.