Data Integration with Talend Open Studio

Robert A. Nisbet, Ph.D.

Most college courses in statistical analysis and data mining are focus on the mathematical techniques for analyzing data structures, rather than the practical steps necessary to create them. But in the business world, the vast majority of situations suitable for data mining require data that are scattered in many different files, formats, databases, and tables. Complex procedures must be designed to perform the Extract, Transform, and Load (ETL) functions necessary to bring this data from various sources systems and formats into a common data structure suitable for data mining and statistical analysis.

Extract

Many data mining projects require data that are housed in normalized data warehouses. Other data mining projects must access data from a variety of sources, each different systems stored in different formats and organizations. Some data may be extracted from the web through spidering or screen-scraping. Other data sources can be demographic databases and files. In any case, data must be extracted from source systems in a denormalized organization to create an analytical file composed on one record per entity (e.g. customer) associated with a number of attributes associated as fields (columns) in the record.

Transform

The transform operation applies a set of rules or functions to the extracted data to change the data to conform to the required format, or to derive new factors (variables) for use in data mining. A detailed explanation of some transform operations commonly performed in data mining is presented below.

Load

The load operation inserts the processed data into some target data structure, usually a data warehouse. Processing for data warehouses augmentation involves loading the data into the schema of the warehouse (normalized relational, Star Schema, or multi-dimensional). Data must be added to a data warehouse in the appropriate manner to be consistent with database operations. Sometimes, data must be inserted in the sorted data structure. Other times, input data must be used to update existing data elements. The load capabilities of the ETL tool provide this flexibility. For data mining, the load process consists of adding model predictions to the data structure for future comparison and refinement of models. Few data mining tools provide sufficient flexibility for loading data warehouses.

Usually, these operations must be performed by dedicated data architect people, using complex and very expensive data integration tools. Data miners who need data elements suitable for data mining must work closely with data architects, or do it themselves. Data warehousing was the first application area where these data integration operations were performed on a large-scale. Very large data warehouses were built by companies like IBM

and NCR to holding many terabytes of data in a highly organized structure suitable for efficient storage and retrieval of business information.

The classical approach for performing these data integration operations was to write SQL programs. A more modern approach was to use SAS, both for data integration and analytical modeling operations. Even newer tools employ a menu-driven graphical user interface (GUI) to orchestrate various data integration operations to create data structures in a format suitable for data mining. The latest generation of data integration tools use a graphical programming user interface to built data flow diagrams composed of icons (components) configurable to conduct specific data operations, which are connected with arrows to show data flow pathways. These graphical representations are translated by the tool into lower-level operations and run to create the necessary output data structures. Examples of these tools include:

1. Informatica
2. Ab Initio
3. Cognos
4. Datastage
5. Oracle Data Integrator
6. Business Objects Data Integrator
7. SQL Server Data Integration Services
8. SAS Dataflux

These commercial ETL tools are complex and very expensive. But, there are some Open Source ETL frameworks available, including:

1. Apatar
2. Clover ETL
3. Pentaho Project
4. Talend Open Studio

The proper application of ETL tools for data warehousing can become very complex, and is outside the scope of data mining requirements. Most data mining tools provide some capabilities for extraction of data from databases, Excel spreadsheets, or flat files. The loading operations of ETL and not very important in data mining, but the transformation processes are extremely important. Data transformation includes all operations necessary to prepare data set to be submitted to data mining algorithms.

Transformation and Data Preparation

Transformation applies a series of rules or functions to the extracted data from the source to derive the data for loading into the end target. Some data sources will require very little or even no manipulation of data. In other cases, one or more of the following transformation types may be required to meet the business and technical needs of the target database:

Table 1 lists some common transform operations, and an annotation listing:

- ETL tool only – these operations are usually performed only by ETL tools, although some data mining tools provide some form of these operation.
- Some DM tools – these operations are supported by some data mining tools
- Both – data mining and ETL tools share these capabilities.

Operation		**Tool**
1. Selection of some or all columns in a data stream		ETL tools only
2. Relating fields of data sources with fields in different orders (mapping) based on the contents of a specified (key) field		ETL tool only
3. Joining data from multiple sources (including lookup and merge operations)		ETL tools only
4. Applying any form of simple or complex data validation. If validation fails, it may result in a full, partial or no rejection of the data, and thus none, some or all the data is handed over to the next step, depending on the rule design and exception handling. Many of the above transformations may result in exceptions, for example, when a code translation parses an unknown code in the extracted data		ETL tools only
5. Translating coded values (*e.g.*, if the source system stores 1 for male and 2 for female, but the source system stores M for male and F for female).		Both
6. Encoding free-form values (*e.g.*, mapping "Ms" title to females) 7. Encoding free-form values (*e.g.*, mapping "Ms" title to females)		Both Both

8. Deriving a new calculated value (*e.g.*, sale_amount = qty * unit_price)		Both
9. Sorting		Both
10. Filtering		Both
11. Deriving a new calculated value (*e.g.*, sale_amount = qty * unit_price)		Both
12. Aggregation (for example, rollup - summarizing multiple rows of data - total sales for each store, and for each region, etc.)		Some DM tools
13. Transposing or pivoting (turning columns into rows or vice-versa)		Some DM tools
14. Splitting a column into multiple columns		Some DM tools

Table 1. Common transformation operations performed by ETL tools only, some data mining tools, and both ETL and data mining tools

Those operations uniquely provided by ETL and data quality tools are provided in the Talend Open System by two processing components:

- *tJoin* – Used to add fields (columns) to existing records based on common contents of a specified field.

- *tMap* – Used to relate fields in two data sources that appear in different orders.

While most of the ETL functions listed above can be performed by most data mining tools, the complex mapping and joining functions needed often to support data mining are not available in data mining tools.

Input Data Sets and Schemas

Before using the tJoin and tMap operations in Talend Open Studio, it is most convenient to load the input data schemas into the *Repository*. The Repository maintains a list of input and output record structures (schemas) that can be referenced and maintained globally in the system. For this tutorial, two data sets are used as inputs, Data1 and Data2. These data sets contain

telecommunications data for customers stored in a different set of fields (different schemas). The initial challenge is to combine all of the fields for a give customer into one record (row) in the output data structure (a file or database table). This operation is performed by *joining* the fields of one input data structure with those of another data structure. Using the Talend Open System to prepare your data sets will accommodate a large variety of data manipulation operations unavailable in the data mining tool.

Joining Operations

There are many kinds of joins supported by ETL tools. Consider the Venn diagram shown in Figure 1 below, consisting of two overlapping circles, representing matching and non-matching records of two data structures. Various types of joins are defined by which different portions of the file coverage are selected.

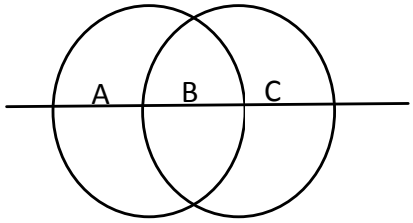

Figure 1. Venn diagrams illustrating areas of matching and non-matching records (according to some key field) for a left-join (A B), a right-join (B + C), and an inner join to include only matching records (B).

The *tJoin* Component of Talend Open System

For many data mining jobs, multiple data sources must be combined for a given entity (e.g. customer) to produce a single row in a data structure containing all fields of the input data sources. Joining two files consists of combining the fields (columns) of one data source with the fields of another data source whenever a key field matches (e.g. Customer_Number). Joining is accomplished by the *tJoin* component in Talend Open Studio. Figure 2 shows a sample join job designed to combine two data sources.

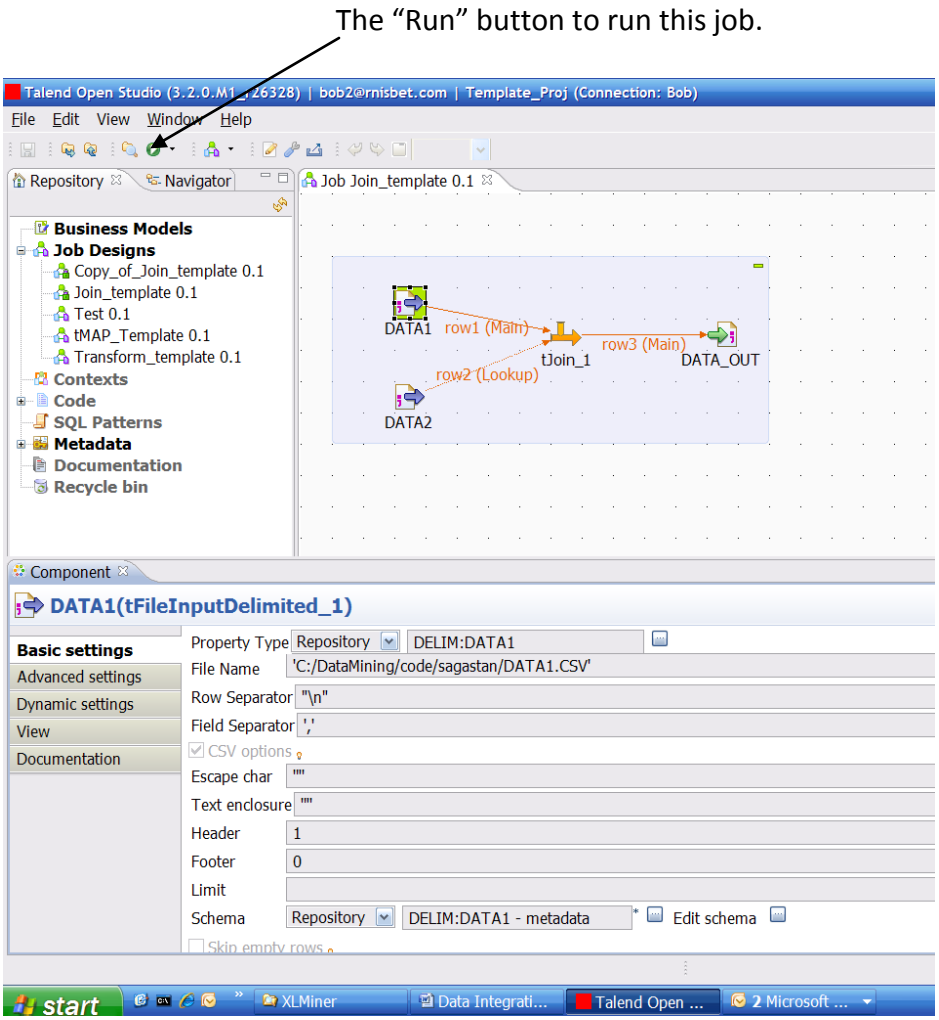

Figure 2. A sample join operation using the tJoin component.

Each data source must be configured to access the appropriate data set according to the requested information in the Basic Settings pane at the bottom of Figure 2.

To see the join conditions, double-click on the *tJoin* component to open the dialog box shown in Figure 3.

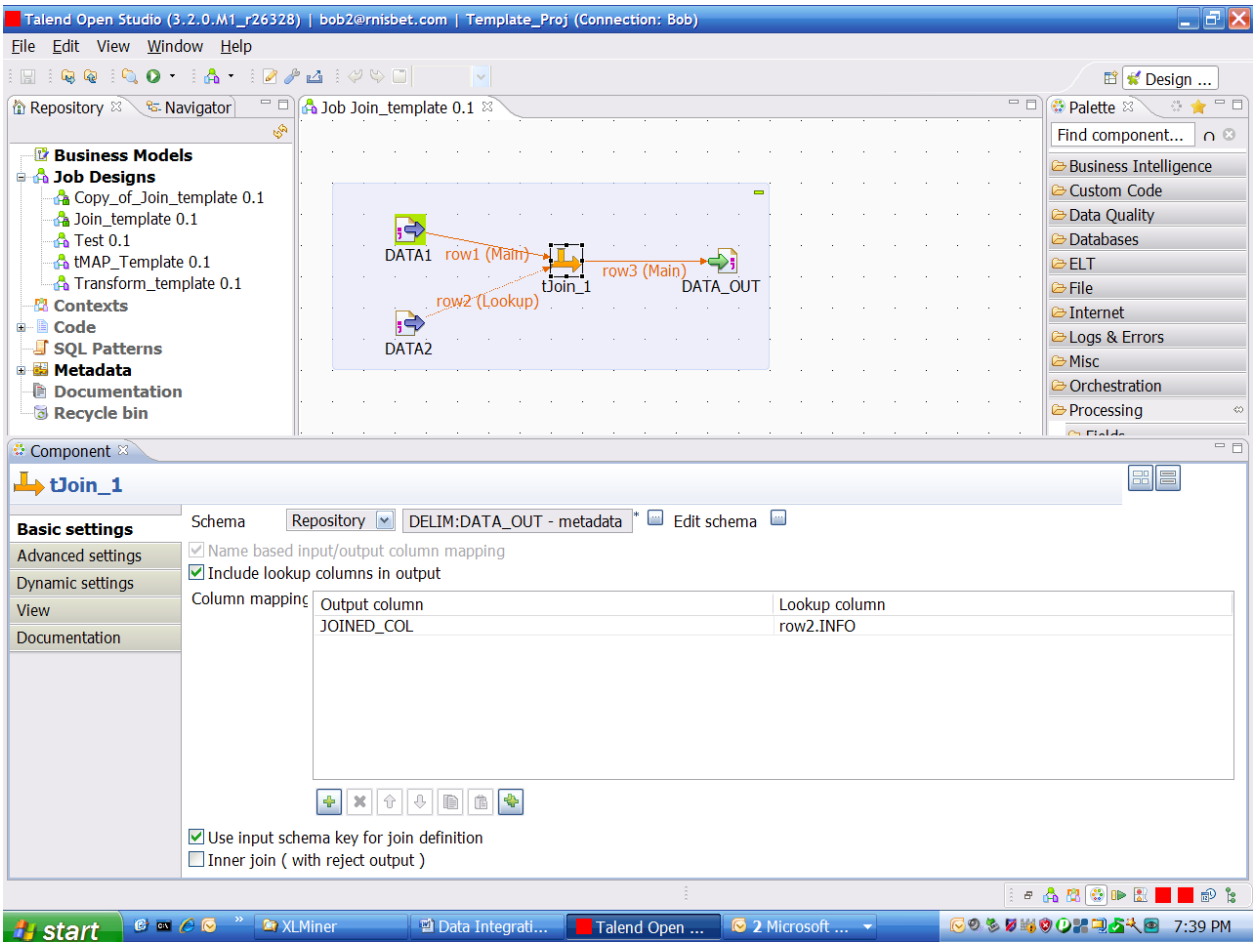

Figure 3. The tJoin dialog box, showing options to specify an inner join and to edit the schema of the join.

The inner join type is specified clicking the inner join option on the screen shown in Figure 3 above. A left join is specified by leaving the inner join box open, and connecting the arrow from the "left" data source (Data1) to the *tJoin* component first, then connect the other data source (which will be treated as the "right" data source). The right data source will be treated as a look-up data source. A right join is not specified explicitly. If you want to perform a join for areas B and C in Figure 1 (which constitutes a logical right join), just connect the Data2 source first. The common field in each data source (the key) is specified in the definition of the schema in the Repository.

To change the default mapping of input columns, click on the Edit schema box in the *tJoin* dialog box to display the mapping schema shown in Figure 4 (select View Schema at the prompt, and click OK).

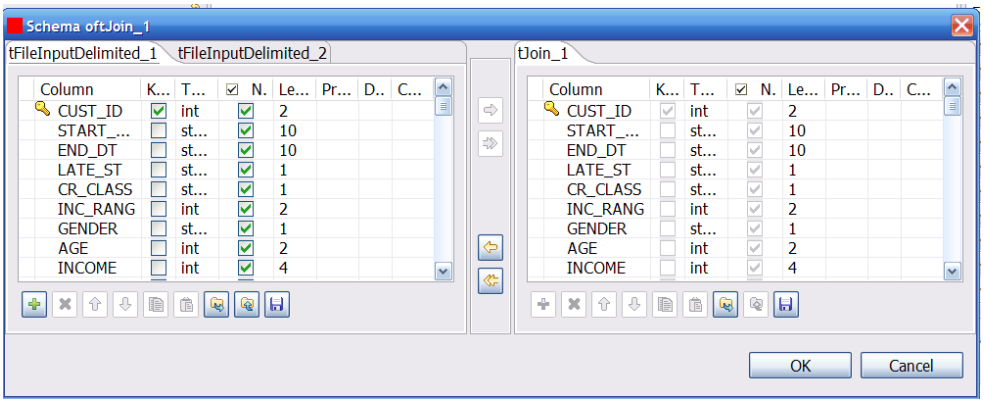

Figure 4. The *tJoin* field mapping schema for data source #1.

Figure 5 shows the field mapping screen for data source #2.

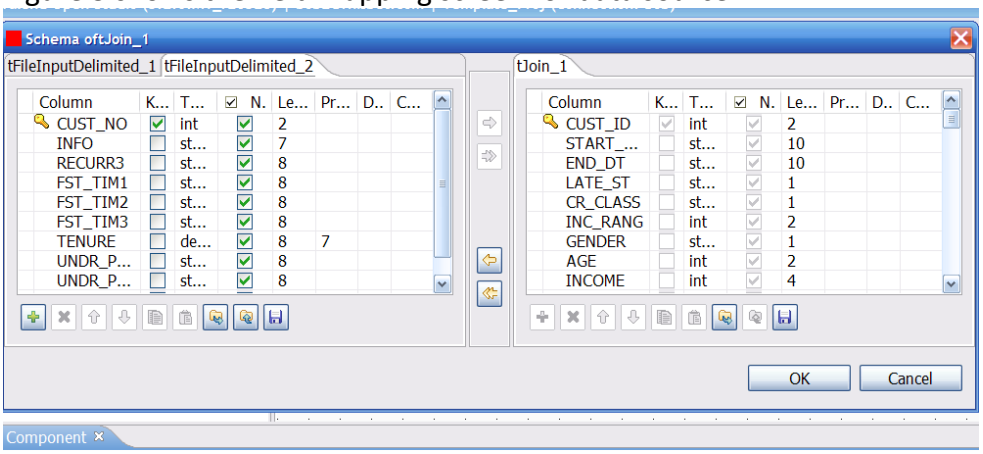

Figure 5. The *tJoin* field mapping schema for data source #2.

The goal of the join operation is to add the fields from data source #1 to those of data source #2, resulting in a record for a given customer (CUST_NO) with all the fields included.

The left pane shows fields for the two input data sources (selectable by clicking on the appropriate tab). The right pane displays the selected fields from both files. By default, all fields from each file are selected. You can remove any input field from the output data structure by highlighting it and clicking the left arrow in the middle of the display at the bottom.

The output components shown in Figures 2 and 3 (DATA_OUT) is a comma-delimited text file. Double-clicking on the component will permit configuration of the component to output the joined data stream in the folder of your choice.

When the configuration of all of the components in the job is complete, you can click on the "Run" button (indicated in Figure 3) to run the job. The resulting joined file is saved with the name and location specified in the DATA_OUT text file output component.

The tMap Component of Talend Open System

Figure 6 shows a job that uses the *tMap* component to map which input fields relate to each field in the output data structure. Fields can be repositioned in the data records with this component.

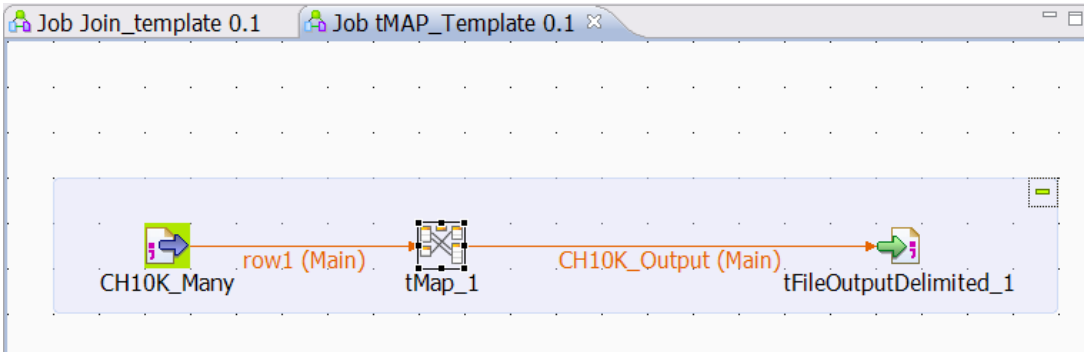

Figure 6. The Talend *tMap* component for mapping fields to an output data structure.

Double-clicking on the tMap_1 component displays the *tMap* configuration screen below.

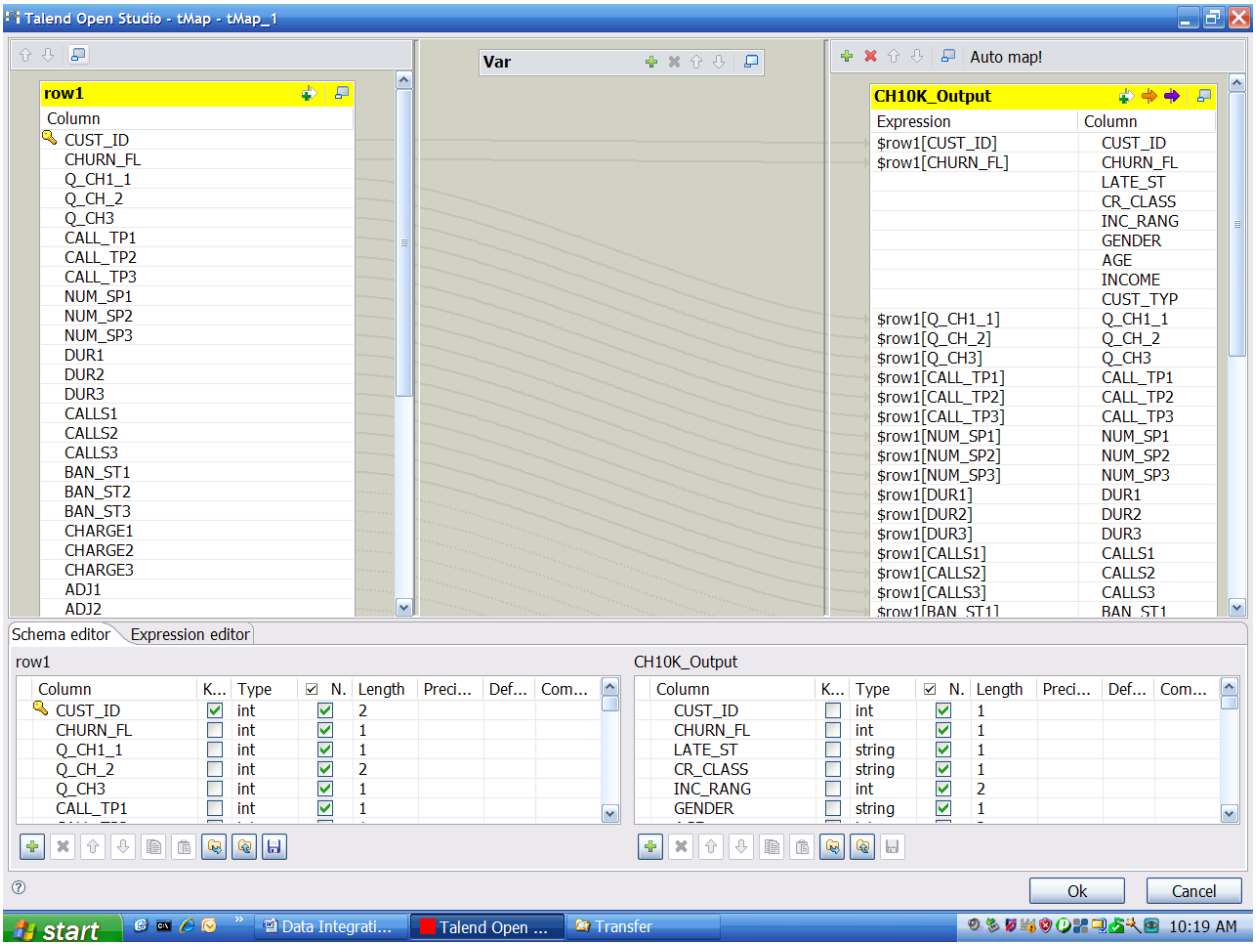

Figure 7. The tMap configuration screen.

The upper display panes show the fields of the input and output schema connected with lines, which indicate the desired mapping. Notice that field #3 (Q_CH1_1) in the left pane is mapped to the variable of the same name as field #10 in the output data structures. The *tMap t*abs located between the upper and lower panes includes a tab for displaying the Expression Editor. Complex transformations of variables can be performed, using either Perl or Java code (configurable upon installation).

The combination of the *tJoin* and *tMap* components permits you to perform many complex data manipulation operations not supported by data mining tools. The combination of the data mining tool and Talend Open Studio can enable the processing of all of the steps in a data mining project, following one of the standard process models (e.g. CRISP-DM). The following discussion of the CRISP-DM data mining process model is found in Nisbet, et al. (2009).

CRISP-DM

This format for expressing the data mining process is the most complete available. It was created by a consortium of NCR, ISL (creators of IBM Predictive Analytics Workbench (formerly, Clementine), and Daimler-Benz companies. The process defines a hierarchy consisting of major phases, generic tasks, specialized tasks, and process instances. The major phases are related in Figure 8 below as it is applied to fraud modeling.

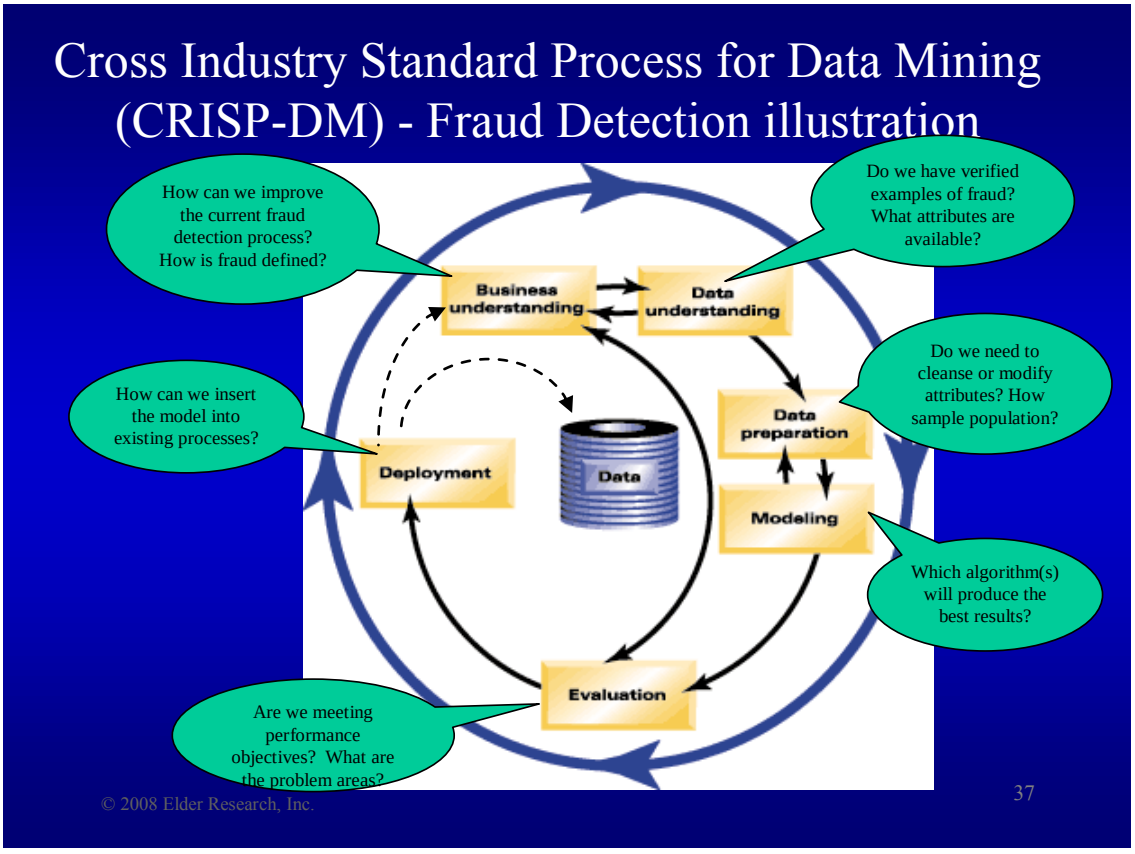

Figure 8. Phases of the CRISP-DM process. The dashed arrows are added to indicate additional data flow pathways necessary to update the database and business understanding.

The dashed arrows shown in Figure 8 represent the closing of the "loop", enriching the Business Understanding and adding predictions to the data base. Each phase of the process consists of a number of second level generic activities, each with several specialized operations. A fourth level (Tasks) could be defined in this process, but these tasks are very domain-specific, that is they must be defined in terms of the specific business problem to be solved in the context of the specific data used to solve it. See Nisbet, et al. (2009) for a detailed description of the CRISP-DM process.

References

Nisbet, R, J. Elder, and G. Miner. 2009. The Handbook of Statistical Analysis & Data Mining Applications. Academic Press (Elsevier). Burlington, MA.