Carleton

The Four Challenges of Customer-Centric Data Warehousing

TABLE OF CONTENTS

Introduction

With today's ever-increasing competitive pressures, higher customer expectations, and new enabling technologies, customer-centric data warehousing is increasingly becoming an important business opportunity. Yet such customer-focused initiatives have – at least historically – been difficult to implement. This paper will look at the risks and rewards of customer-centric data warehousing, and will specifically focus on four key technical challenges unique to data warehouses that center on customer data. This paper addresses each of these four challenges and shows how — as never before — the benefits of customer-centric data warehousing can be attained today.

Customer-Centric Data Warehousing - What is it? What are the rewards?

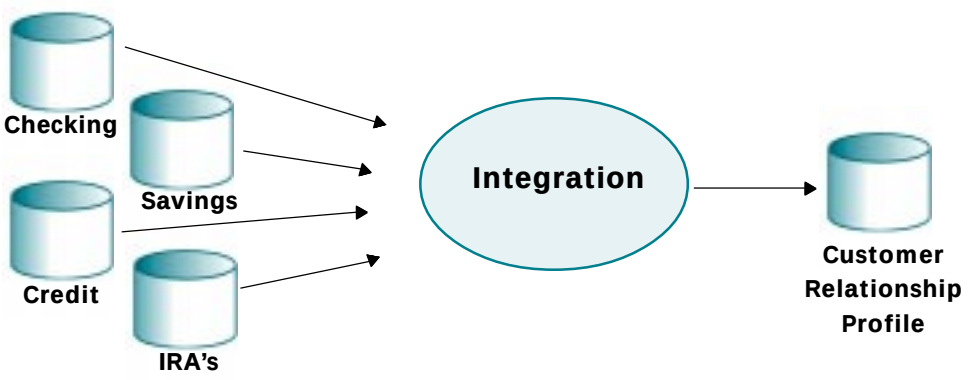
Customer-centric data warehouses are simply data warehouses that require complete, accurate views of customers, with their associated data, to solve important business problems. We are using *customer* broadly here to mean individual customers, business customers, households, prospective customers, and even vendors and suppliers. While the concept is simple, implementation of customer-centric data warehouses is often easier said than done – as we shall see.

Integrating Customer Data Throughout the Enterprise

The value of the customer-centric data warehouse or data mart is due to its integration, or consolidation, of customer data. The warehouse integrates customer data that is fragmented across multiple sources within your organization. The sources often include various business line systems that support the following functions:

- Order processing
- Customer support
- Inquiry systems
- Marketing
- Various transaction systems, etc.

Businesses' transaction systems are typically designed to deal only with the needs of that transaction process. Therefore, each transaction system has a piece of information about the customer. For example, a bank will typically have these transaction systems:



Data in these transaction systems are often organized around "accounts" or "policies" or other similar transactional concepts, that limit the ability to identify unique customers and their total relationship to the business.

In the customer-centric data warehouse, the customer data from each transaction system will be brought together, or integrated, to provide a whole, unified view of the customer.

Customer-Centric Data Warehouse Applications

In the customer-centric data warehouse, the customer data from each transaction system will be brought together, or integrated, to provide a whole, unified view of the customer. There is tremendous value to businesses in possessing this unified view of customers. Here are just a few examples of the many ways in which integrated customer data may be used to achieve competitive advantage, improve customer service, or reduce operations costs:

- Customer relationship management
- Consolidating customers after acquisitions or mergers
- Greater revenue and profitability from customer retention, cross-selling, and cost management
- Fraud detection
- Reliable data mining results
- Vendor consolidation
- Healthcare Master Patient Index (MPI)
- Healthcare outcomes analysis
- Customer service

External Supplementary Data and Householding

Building a unified customer view may even involve integrating data from outside your organization, such as demographic data, or credit information, to enhance your customer knowledge. Further, a customer view often involves establishing relationships, often referred to as *householding*. Householding is identifying the individuals who are in the same family, or household, often for marketing purposes. Similarly, business householding is identifying the relationships within a business, such as all those individuals at a particular location of a larger organization.

Building Customer-Centric Data Warehouses Is Challenging

The business rewards for integrating and using customer data are enormous. But customer-centric data warehouses, until recently, have been difficult to build. Why is that so? First, building *any* data warehouse or mart is challenging. Numerous articles have been written on the benefits and challenges of data warehousing and there's no need to repeat that. Rather, we will focus here on what is particularly challenging about customer-centric data warehouses.

The Four Unique Challenges

There are four reasons in particular that customer-centric data warehouses can be difficult to build and maintain:

1. **Customer data requires integration**. Rarely is a reliable common key available on which to match up and identify the same customer from different sources, and then merge the data.

2. **Names and addresses are difficult.** Usually, the matching must include matching on names and addresses, which requires specialized tools (and often specialized knowledge)

3. **Update requirements are complex.** Customer data warehouses must typically be updated, rather than completely refreshed, or updated in increments (such as adding another week of retail sales). This updating is more complex than a refresh or incremental addition.

4. **Requires separate transformation and cleansing tools.** Historically, there have not been tools that provide a complete solution for building customer data warehouses.

Let's understand each of these challenges and, most important, how to address them.

Challenge 1: Customer Data Requires Integration to Build a Unified View

As we've seen, a customer-centric data warehouse must combine customer data from internal sources, and may require adding external data. Unique customers must be identified across and within all these sources. And the business needs may require not only identifying the individual customers, but also the relationship between customers, such as household relationships. Let's look at the matching and merging capability required to integrate this customer data.

Fuzzy Matching Required

Since it is very rare that all sources have the same customer identifier (primary key) matching customers across sources is difficult.

Since it is very rare that all sources have the same customer identifier (primary key), matching customers across sources is difficult. It requires what is often called *fuzzy matching*. Fuzzy matching uses algorithms to identify similar records (rows). It is probabilistic, and only records that have a very high confidence of a match should be considered a match.

Several columns of data that in some way identify a customer, that are in common across the systems, are used for matching. For example, fuzzy matches might be attempted on these three columns:

- Phone number,
- Name, and
- Address

Typically, matches are attempted in several different ways, in accordance to a business' rules for what they consider a match. For example, in addition to the above three columns, in order to "catch" individuals who have moved and therefore have different addresses, one might also match on:

- Credit card number,
- Date of birth, and
- Address

Two records are considered a match when a match was made on either set of data and business rules.

Match Candidates Must Be Clustered

One complicating factor in matching, is that the number of match comparisons to be attempted grows exponentially with the size of the data, if every record is compared with every other record. In fact, the formula is:

$$(n^2 - n) / 2$$

With even a small number of records, let's say 100,000, the number of comparisons would then be 4,999,950,000! (almost 5 trillion!)

Therefore, records must be clustered into candidate groups, or *workunits*. The idea is that only records within a workunit are compared with each other. This dramatically improves performance. Careful selection and management of workunits is needed to avoid missing matches.

Matched Data Must Be Merged (Consolidated or Integrated)

Once a match is made between two records (or three or four, for that matter), the records must be consolidated, or merged. After all, that's the whole reason the data warehouse is being built – to integrate the important data about the customer. Different information will be merged from each source, for example:

- Current balances
- Credit status
- Investment activity
- Support contacts
- Direct marketing response

Sometimes the information to be merged, such as personal identification information, is in more than one source. Then, merge rules determine where the data is merged from. For example, a field might be merged by:

- Source priority (e.g., the checking address is considered more reliable than the IRA address)
- Most recent update (the source with the most current date stamp will be used)
- Most frequently occurring (e.g., three of four sources have the same phone number)

According to the META group, healthcare organizations implementing master patient indexes have found 5 to 30% of their patient records to be duplicates.

Imperative to Business That the Match/Merge Be Done Right

It's important that the match and merge be done as completely and correctly as possible. When matches are missed, a single customer will appear as two or more customers. You will have an incomplete view of your customers, and your customer count will be exaggerated. According to the META group, healthcare organizations implementing master patient indexes have found 5 to 30% of their patient records to be duplicates. Even worse, if two individuals or companies are matched that shouldn't be, you get a warped view of the customer. Customer matches that are incomplete, inaccurate, and unreliable thereby lead to incorrect business analysis and poor decisions. These decisions can cost the organization losses in revenue and higher expenses, just the opposite of what the customer data warehouse was intended to achieve. It is very important that you or the project sponsors identify and estimate the cost of each incorrect customer match or missed match, in terms of lost revenue, lost profit, or unrealized cost savings.

Examples include:

- Assigning the wrong accounts to a customer and wasting valuable marketing resources and customer goodwill in cross-selling products this customer already has
- Losing a profitable customer because the customer service representative doesn't understand the relationship value
- Targeting prospects based on wrong marketing criteria
- Not recognizing the full credit risk associated with the customer

Challenge 2: Names and Addresses Require Special Attention to Match Customers

Names and addresses are very important to the accurate matching of customers and building a complete, accurate customer data warehouse, for several reasons:

- Unique IDs, SSN, Date of Birth, etc. are often unavailable or inadequate
- Names and Addresses are almost universally available

For effective customer matching, names and addresses must be cleansed. Now let's look at each of these points regarding names and addresses, and customer matching.

Non-Name and Address Data Not Adequate for Matching

There are several highly desirable fields to help identify customers, such as social security number, phone number, date of birth , and DUNS number (Dun & Bradstreet identifier) that are often not available. If the data is available, it is usually available for only some sources or some percentage of records within a source. Furthermore, when the data is available, the data may not be reliable. These are some examples. With area codes bursting at the seams, new phone numbers are too quickly re-assigned to new households. A parent or trustee's social security number is used in place of a child's. Typos of various sorts occur. And the stories of matches made on SSN alone, with thousands of matching 999-99-9999's (or similar "placeholders"), are legendary.

Name and Addresses are Ubiquitous but Messy

Names and addresses, on the other hand, are typically in almost every source and record. And they both have high cardinality, making them excellent personal identifiers. That's the good news. The bad news is that names and addresses are the most difficult type of customer data to work with. Invariably, name and address data must be cleansed in order to be used for matching. Unlike, say, date of birth, there is tremendous variability that can occur even between what are actually the same name and address. Here are examples of how the same ***individual's*** name and address can easily vary:

Invariably, name and address data must be cleansed in order to be used for matching.

- Nicknames
- Marital status change, and consequent change or hyphenation of last name
- Variation due to formality (A.J. Jones and Dr. Andrew Jones PhD)
- Omitting Jr., Sr., II, or III
- Abbreviations in address (E. and East Ave and Avenue)
- Including or not including apartment #
- Changing to or from PO Box rather than street mail delivery
- Spelling errors and typos can be hard to identify (how do you know if it's 100 Main or 1000 Main?)

Similarly, there is tremendous variability in ***company*** name and address data:

- Titles change
- Titles may be entered in name field
- Company names are abbreviated in many ways (PPG, PP&G, PP&G Inc., Pittsburgh Paint & Glass)
- Companies may be identified by division or parent or both (GM, Cadiallac, Cadillac – GM, etc.; the variations are almost limitless)
- Individuals may move frequently within corporate campuses,
- Mailstops, department names, and building identifiers may be mixed in with mailing addresses or company names

Varying Formats Require Parsing

Let's imagine that your data is 100% clean. That is, every instance of the same customer is represented in exactly the same way in the various source databases. All the data is perfectly complete and consistent. Then there would be no need to cleanse the names and addresses, right? Wrong! Invariably, the data from different sources is formatted (fielded) differently. The data from each source must be parsed into consistent fielding. And furthermore, for matching purposes an individual name or a street address, as examples, must be parsed into their finest elements, such as first name, last name, street number, street name, and so on.

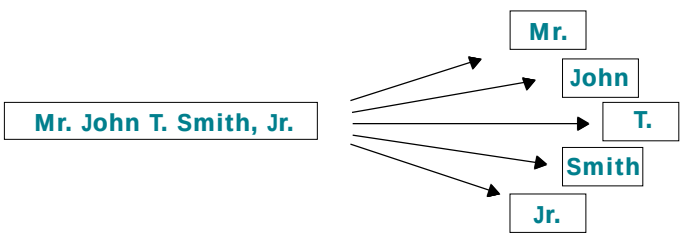


FIGURE I. Name Parsing

Quite often, on close analysis, you will find data has been mis-fielded or mis-entered. The parsing routines of name and address tools can be used to fix this by concatenating fields and re-fielding them from the tools' results.

Capabilities of Typical Name/Address Processing Software

For all these reasons, it requires a special-purpose solution to cleanse name and address data. Name and address processing software typically will:

- *Parse* name and address information into the individual elements for improved correction and matching
- *Standardize* (consistently represent) names and addresses for improved matching
- *Correct* address components, such as street names, city, and zip code for greater accuracy
- *Augment* with new data elements for better target marketing or other purposes
- *Reformat* for merging the data and populating the warehouse

Here are some examples of these above functions, for name and address processing respectively:

Function	Name	Address
Parse	Individual, Company Name, plus title & "firm location" (dept. mailstop, building name, etc.)	Mailing addresses
Standardize	Doctor to Dr, Vice Pres to VP, Bill, Billy, etc. to William, etc.	Street or Str to St. East Main to E Main
Correct	NOT APPLICABLE to names	Street names, city names, ZIP codes
Augment	Add gender codes	Add ZIP4, county codes, etc.

Name/Address Pre-Processing Has a Huge Impact on Accurately Identifying Customers

This parsing and cleansing of name and address data *dramatically* improves the quality of matching, resulting in:

- Fewer missed matches
- Fewer erroneous matches

When matching on names and addresses, specialized processing tools are an absolute must, and will contribute more to quality matching, and data warehouse quality, than the most sophisticated matching algorithms and rules.

Challenge 3: Update Complexity to Maintain the Unified Customer View and Add New Customers

The maintenance of customer data warehouses is different, and more challenging, than the maintenance of most other types of data warehouses. Customer data warehouses must be incrementally updated, rather than completely refreshed (re-loaded). Let's look at why this is so, and what the implications are.

Customer Data Warehouses Require Incremental Updating

Why is customer data usually an update situation rather than a complete batch refresh? A customer warehouse or mart contains all customers (or vendors, prospects, etc.) depending on the purpose of the warehouse or mart, and generally a long history of data. This means that totally rebuilding the warehouse at each update period is usually not an option, for two reasons:

- The historical data is no longer available on the source systems.
- The amount of processing to totally re-match and rebuild each time you update is not feasible within the batch update window (overnight or over the weekend).

Since it is not practical to re-build and refresh with each update period, the customer data warehouse must be incrementally updated.

Therefore, since it is not practical to re-build and refresh with each update period, the customer data warehouse must be incrementally updated. This means that new data from each update period must be added to the data warehouse, while preserving existing data. More specifically:

- New data must be identified as belonging to a new or existing customer
- New customers must be given a unique ID, and new rows inserted
- Existing customers must have their data updated, on a column-by-column basis (to reflect additional purchases, change in service status, or whatever other data is stored in the data warehouse about the customer and their activity)

Contrast the update requirements of a customer data warehouse to that of a warehouse of retail sales data, containing no customer information. Each week new sales data is added to the warehouse. But there is no need to match incoming data with existing data. There is no change to the existing detail data. There are no field by field incremental updates. It's true that some of the data is also rolled up by sales season and by day of week, for example. But that is typically handled as a complete refresh with the rolling up occurring before getting into the database. So, the maintenance of this type of database involves a complete re-load. This is significantly simpler to maintain than the updating of the customer database.

Incremental Updating is More Complex Than Refreshing and Often Poorly Supported

From the above description of what needs to be done with a periodic update of a customer warehouse, we can derive that the system or tool doing the update must have these capabilities:

- Match incoming (source) data with warehouse data to distinguish whether the incoming customer is new or existing.
- Create Unique IDs for new customers
- Conditionally insert
- SQL update on a column-by-column basis
- Updates may involve calculations using values in existing columns (for example, total sales history for an individual, or total value of all accounts)

The first point, matching incoming with existing warehouse data, deserves a closer look. To understand the issue, let's consider an example where the data warehouse contains 10 million customers, and it is updated nightly with information on an average of five thousand customers. Some tools and systems will require you to unload all the customer identification information tables in order to match up the incoming and existing customers. With batch update window time constraints, and matching being an inherently processing-intensive activity, this is usually not feasible. Instead of unloading all the data, only true candidates for matching should be read from the warehouse. This dramatically reduces the processing time required. This type of periodic updating and matching is called *synchronization*. It *synchronizes* the incoming and existing data efficiently without totally re-doing the initial processing. Because of architectural constraints or relational I/O limitations, many tools cannot synchronize.

As you can see, customer data warehouses have some inherent complexity in the nature of their updating that many other types of data warehouses do not.

Challenge 4: Separate Tools to Build and Maintain the Customer Warehouse

Data Prep Processes in Customer-Centric Data Warehouses

Before looking at the tools that have been available to prepare the data, let's first look at the data-preparation processes needed to build and maintain a customer-centric data warehouse. From our discussion of the special challenges of customer data warehouses, we know customer-centric warehouses have several processes not usually encountered in data warehousing data preparation:

- Match & merge
- Name and address processing
- Synchronization and incremental updating

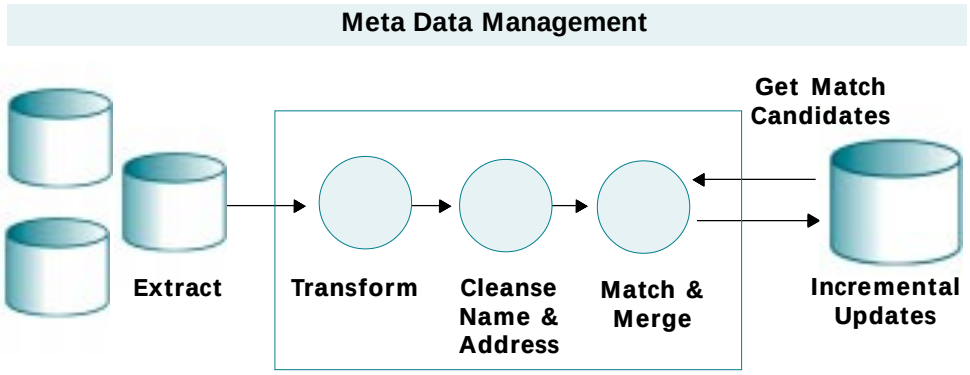


FIGURE II. Functional Requirements for Customer Data Warehouse Preparation

In the above diagram, see how the name and address cleansing is a necessary precursor to matching. Note how not only do we have to process the source data, we also need to get candidates of possible existing customers to match against. When the match and merge is complete, then updating is done (not a load). Keep in mind that sub-processes are shown to help you conceptualize the process. The sub-processes are not separate batch processes, involving files to disk. Rather, they should actually be done in memory, for performance.

Two Types of Tools Needed in the Customer Data Warehouse

Now that we have an overview of the processes needed, let's look at the two types of tools that are used to prepare the data in customer-centric data warehouses:

- Data warehouse *transformation tools*
- Cleansing products

We will see that though there has been some confusion about these tools, because there is some overlap in functionality, they are for the most part quite different. Both are needed to build and maintain a customer data warehouse.

Transformation Tools Meet Generic Data Warehousing Needs Not Special Needs

Over the past several years, transformation tools have become widely used to build and maintain the data warehouse. The primary functions of these tools are shown in the schematic below:

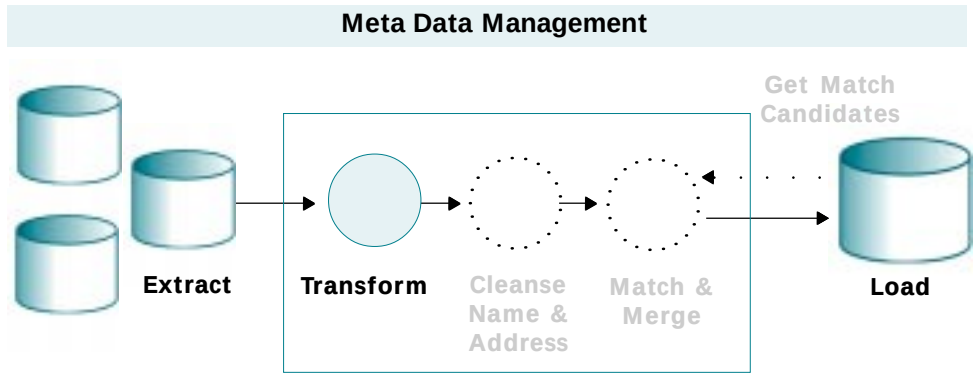


FIGURE III. Capabilities of Transformation Tools Incomplete

Note how these tools don't include all the processes that we identified as necessary: They *lack*:

- Name and address cleansing
- Fuzzy matching and merging

And, typically, their synchronization with the target database and update capability is weak. The reason for this is that these transformation tools, with one exception, were not designed for customer integration. They were designed for simpler extraction, aggregation, and loading. These tools are oriented to OLAP analysis of summarized data, not granular customer-level data.

Cleansing Products Meet Special Needs but Don't Provide General Data Warehousing Needs

Cleansing products, as the name implies, primarily cleanse and match. All of the most commonly used products focus on names and addresses. They can perform the name and address processing previously described, to some extent. And they perform the matching and merging described earlier, again to varying degrees.

On the other hand, they *do not provide*:

- Extraction
- Load or update
- Metadata management

Where they overlap with transformation tools, is in their transform/validation capability. The cleansing tools vary widely in this regard. They vary, with non-name and address data, from no capability to very extensive data discovery capability beyond what some transformation tools can provide. But mostly, the core focus of the cleansing tools is just that, cleansing and matching.

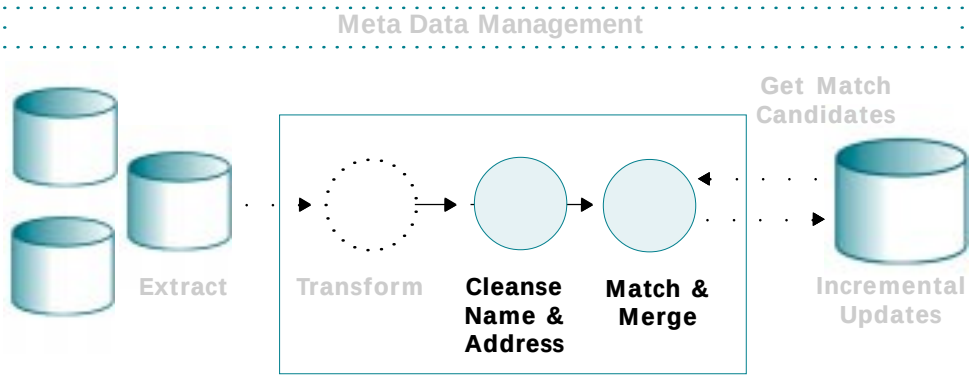


FIGURE IV. Capabilities of Cleansing Tools Incomplete

The cleansing tool's name-and-address focus is easy to understand when you understand the origins of these products. Whereas transformation tools were designed specifically for data warehousing, the origins of cleansing products are:

- Mailing/direct marketing
- Mainframe customer consolidations (acquisitions/mergers) or customer application conversions, such as in banking

Consequently, all of these products are flat file in/out oriented, though some are available as I/O-less libraries. They do not have RDBMS read/write capability. They didn't need this since mailing services bureaus work on a tape in/out basis. And mainframe system conversions are batch processing oriented. Furthermore the mainframe conversion products are written in COBOL and thus not well suited to the open-systems oriented data warehousing tasks.

Not All the Cleansing Products Provide the Same Functionality

The mailing/ direct marketing products tend to be especially strong in the address correction, whereas the consolidation/conversion products lack the ability to correct or validate street-level data because they lack the United States Postal Service address database and do not go through the accuracy certification process of the USPS. All these products tend to work well with cleansing names.

The other variation in the cleansing products is their sophistication with non-name and address data and their ability to merge. Again, this is due to their heritage. Since in mailing the only data typically handled is names and addresses, sometimes the products have limited matching flexibility and limited ability to work with a variety of data that can be identifying (Date of birth, SSN, credit card, account numbers,etc.). Their field-by-field merge capability tends to be even more limited. This is often not realized, in part because in mailing these products are called *merge/purge* products, which makes it sound as though they're designed to merge. However, merge – in mailing – means something very different. In mailing, merging means that matching records from various mailing lists are purged out except for the one survivor to be mailed, whereas in data warehousing data from the matching records are merged together. Furthermore there ability to carry along or work with large amounts of data (such as sales info) that is not used for identifying, is limited.

To summarize, cleansing tools tend to be strong in processing and matching names and addresses, but -alone- are not geared towards data warehousing...

To summarize the cleansing tools, they tend to be very strong in processing and matching names and addresses, but - alone - are not geared towards data warehousing. This is because they can neither source nor load / update relational databases, nor provide the metadata management environment that is especially important for maintenance, and often lack adequate transformation/mapping functionality.

Customer-Centric Data Warehousing Demands an Integrated Cost-Effective Transformation & Cleansing Tool

As you can see, transformation and cleansing tools overlap slightly, but for the most part are almost completely complementary. Both types of products are needed.

So buy both, right? But there been two problems with that approach:

- The tools are not integrated together.
- Both types of tools are expensive (often $150K - $200K each)

The integration effort for a user is time-consuming and costly. Integration at both the data and metadata level is needed. And often the architectures of the two products do not fit well together into the overall data warehouse architecture. With the integration effort also comes increased project risk.

Since both products are expensive, due to budgets, and to some extent, confusion about overlapping capability, users have sometimes found it hard to get the justification to purchase both. The result has been serious data quality problems, often leading to failure of the data warehouse when cleansing tools were not purchased. Or when transformation tools were not purchased, extensive and expensive hand coding required to fill in the gaps, and resulting in a very difficult and costly system to maintain, also due to poor metadata management.

What is needed is a transformation tool that integrates the features of both transformation and cleansing tools and addresses the unique challenges of customer data warehousing.

What is needed is a transformation tool that integrates the features of both transformation and cleansing tools and addresses the unique challenges of customer data warehousing. Wayne Eckerson, previously of Patricia Seybold and now with The Data Warehousing Institute, has written of exactly the need for the complementary functionality of transformation and cleansing tools to be integrated, and to be provided as a cost-effective product. Such a tool would completely handle the data preparation processes previously described:

	Transformation Tools	Cleansing Companies	Integrated Tools
Meta Data Management	✔		✔
Transformation	✔	limited	✔
Name & address cleansing		✔	✔
Fuzzy matching and merging (integration)	Usually not	✔	✔
Relational I/O (source, read DDL, load, update)	✔		✔

Very recently such an integrated solution has become available. It provides transformation and cleansing in one product. This saves integration cost and effort, reduces licensing costs, reduces risk, and increases the likelihood of project success. This is a significant development in the data warehousing industry and a milestone in dealing with the data quality issues that are so critical to data warehouse success.

A Tool Evaluator's Checklist - What to Look for

Let's summarize the key points we've identified regarding the customer data warehouse challenges as it applies to the tool needs for data preparation. When evaluating such solutions, as we've seen, here are the some checklist requirements:

- An integration of transformation and cleansing capabilities
- Single source for implementation and service of integrated product
- Consulting or implementation services experienced with the challenges to successfully building customer data warehouses
- A more cost-effective solution than buying the two products separately
- An architecture designed for integration – for fuzzy matching and merging
- The ability to synchronize with and intelligently update the data warehouse
- Sufficient flexibility and sophistication in matching and merging to implement your business rules for matching and merging
- "CASS" level data (street correction) in the address processing (in Canada, look for "SERP" data, from Canada Post). Specifically look for CASS data but you do not need generation of a "CASS certificate" as that is needed only for submitting mailings.

Conclusion - Complete, Integrated Transformation and Cleansing Tool Is Essential to Success

Customer-centric data warehouses — data warehouses that integrate customer data — have tremendous business value. We've seen that there are four technical challenges characteristic of customer data warehouses that differ from other types of warehouse projects:

- Customer data requires integration since sources rarely have a common key. Fuzzy matching, and an architecture to enable that matching and merging, is required.
- Usually, the matching includes names and addresses, which requires specialized tools to parse and cleanse the data to facilitate high-quality matching
- Customer data warehouses typically require synchronization and incremental updating, rather than simpler refreshes (re-loads).
- Historically, two tools have been needed to prepare the customer data – transformation and cleansing tools. They provide complementary functionality, both of which are needed. Only recently have integrated cost-effective tools that combine the functionality of both become available.

Integrated customer integration tools for data warehousing simplifies all the challenges, reduces the risks, and increases the potential rewards of customer-centric data warehousing.

Notes

[1] Journal of Data Warehousing, Volume 2, Number 3, Fall 1997, pages 30-32, "Finding the Dividing Line Between Data Cleansing and Data Transformation Tools," Wayne Eckerson, Patricia Seybold Group