

Managing Data Quality

DataMirror[®]
The experience of now.[™]

TABLE OF CONTENTS

<u>THE NEED FOR QUALITY DATA</u>	1
THE DATA PROBLEM: QUANTITY AND COMPLEXITY	1
THE SOLUTION: A CORPORATE INFORMATION SYSTEM	2
THE ROLE OF DATA WAREHOUSING	3
A DATA TRANSFORMATION MANAGEMENT SYSTEM	5
<u>A DATA TRANSFORMATION MANAGEMENT SYSTEM</u>	6
THE CLEANUP AND TRANSFORMATION PROCESS	6
MANAGING A DTMS	8
MANAGING COMPLEXITY	8
VENDOR APPROACHES	9
<u>THE CONSTELLAR HUB</u>	12
OVERVIEW	12
CONSTELLAR OPERATION	13
CONSTELLAR DEVELOPMENT	14
MANAGING CONSTELLAR OPERATIONS	15
<u>SUMMARY</u>	18

THE NEED FOR QUALITY DATA

THE DATA PROBLEM: QUANTITY AND COMPLEXITY

Organizations now generate more data

After building database systems for over thirty years, organizations are now generating and gathering more data than at any other time in corporate history. The staggering growth of the Internet and Web technology is also adding to the mountain of data available to business managers. More data, however, does not necessarily mean better information, or more informed business decisions. Often data accumulates so rapidly that organizations cannot put procedures in place fast enough to ensure data accuracy and quality. For corporate decision making, business users need access to clean and consistent data.

Business processes are more complex

The nature of data is also changing – becoming more complex. There are two reasons for this. The first is that business processes themselves are becoming more complex. To attract clients and boost profits, insurance companies, for example, have been offering an increasing number of products over the years. The same situation is occurring in the banking and telecommunications industries as companies offer more options and add new lines of business. This all adds up to increased complexity. This complexity makes life difficult for the business user who would prefer to look at a single database to find out what business a client does with the company, rather than have to wade through various account files to find the same information.

Mergers and acquisitions also add to the problem

The second driving force behind increasing data complexity is the number of mergers and acquisitions taking place due to today's tough and highly competitive business climate. The merging of companies means that the associated IT systems must also be merged, or at least integrated. Often, an IT department may be in the middle of one integration project when yet another merger or acquisition occurs. Managing and controlling data in such an environment is a nightmare.

Data quality suffers

The key data issue that organizations face today is that data quality, and, therefore, the business decision-making process itself, is being affected by increasing data volumes and complexity. Unless an organization finds a way of solving and managing this issue, data anarchy will result and business users will be unable to manage the business properly because of poor or inconsistent information.

The solution to this problem is to build a corporate information system that provides high quality, consistent information to business users. Such a system can reduce costs, improve profits and enable an organization to compete more effectively. This report looks at the latest techniques for building a corporate information system: focusing specifically on how to transform data into business information. It also reviews Constellar Corporation's Constellar Hub¹ product,

¹ Constellar Corporation was formerly known as the SQL Group. The Constellar Hub was formerly known as Information Junction.

which is designed to support organizations that need to transform large volumes of complex data into useful business information.

THE SOLUTION: A CORPORATE INFORMATION SYSTEM

A corporate information system improves data quality

The objective of a corporate information system is to manage the quality and consistency of information flowing to business users. The components of such a system are shown in Figure 1.

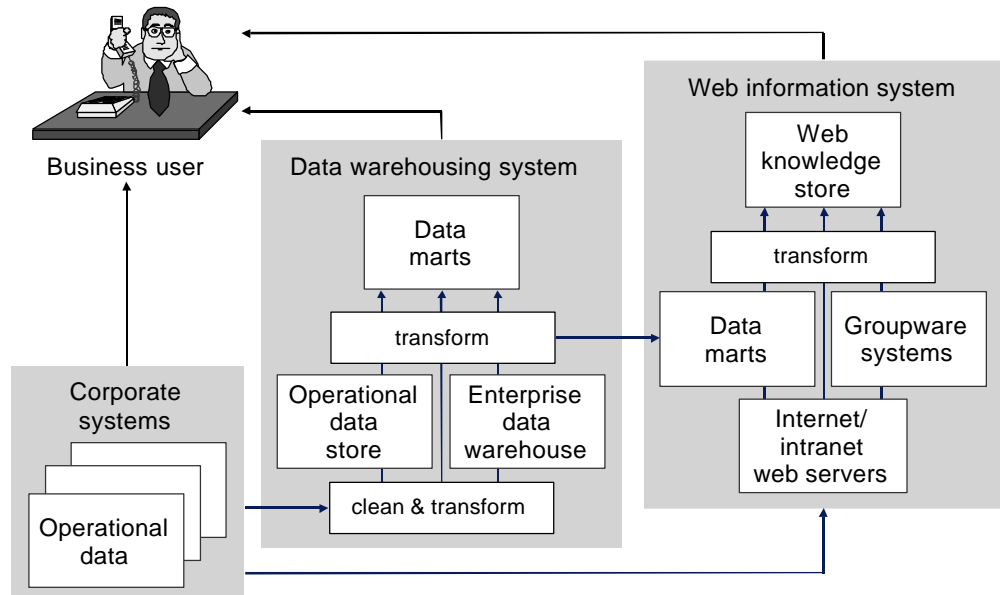


Figure 1. A corporate information system

Fixing data quality problems is expensive

Corporate operational systems manage the day-to-day business operations of an organization: order entry, inventory management, shipping, invoicing, and so forth. Over the years these systems have evolved from employing basic file systems to using a wide variety of different database products such as IDMS, IMS, DB2, Oracle, etc. These systems are designed with performance, rather than the business user, in mind. This is why users have never held many corporate operational systems in high regard; they do not provide the information needed to analyze business operations, or to make business decisions. Acquisitions and mergers have just aggravated the situation, and made even the job of providing an integrated operational system difficult to achieve. Add in Year 2000 problems, and you begin to see why many organizations are being forced to reengineer and downsize their operational systems to reduce spiraling maintenance costs. The Gartner Group estimates that organizations will spend \$400 to \$600 billion converting applications and databases for the year 2000. This is a huge figure for solving what amounts to a basic data quality problem, and is only the tip of the iceberg compared to the rest of the data quality issues that exist within an organization.

One major issue that arises as organizations reengineer and downsize their operational systems is the volume of data that has to be transformed and transported across systems. To be able to do this efficiently organizations need powerful data

transformation tools that can also manage and automate the distribution of transformed data to reengineered and downsized systems.

Data warehouses provide clean and integrated data

A **data warehousing system** helps solve the problems of poor support for business information by operational systems. The objective of a data warehousing system is to capture corporate data from operational systems and clean and transform it into a consistent form that is understandable and has business context for the end user. The key words are “clean and transform.” To quote the often-heard phrase “garbage in garbage out,” unless effort is put into cleaning and transforming the operational data, a data warehouse is a waste of time and money. There are many different types of data warehouse – some are used for tactical decision making, others for making more strategic decisions – these various types are discussed in the next section “The Role of Data Warehousing.”

Extending the concept of a data warehouse to the Web

A **Web information system** integrates data from operational, data warehousing and groupware systems (such as Lotus Notes and Microsoft Office) with information stored on Web servers on a corporate intranet and the public Internet. A Web information system extends the notion of a data warehousing system to include all types of information of interest to a business user, making it accessible from Web-based desktop and network computers.

THE ROLE OF DATA WAREHOUSING

Integrated information for decision making

A data warehousing system supplies business users with clean and consistent corporate information for strategic and long-term business planning, and for making day-to-day tactical business decisions. This information may be used for managing business processes affecting a specific business unit, multiple business units, or the whole enterprise.

Many different warehouse configurations are being deployed by organizations today. Most contain one or more of the types of warehouse database shown in Figure 2.

Data warehouse databases vary in the type of data they manage (detailed vs. summarized, historical vs. current) and the scope of the business issues they address (single business unit vs. multiple business units).

An EDW contains historical data

An **enterprise data warehouse (EDW)** contains detailed (and possibly summarized) data captured from one or more operational systems, cleaned, transformed, integrated and loaded into a separate subject-oriented database. As data flows from an operational system into an EDW, it does not replace existing data in the EDW, but accumulates it to show a historical record of business operations over a period of time that may range from a few months to many years. The historical nature of the data in an EDW supports detailed analysis of business trends over a period of time, and this style of warehouse is used for short- and long-term business planning and decision making covering multiple business units.

An ODS reflects the current status of operational systems

An **operational data store (ODS)** presents a subject-oriented, integrated and consistent picture of the current data stored in operational databases. As data is modified in operational systems, a copy of the changed data flows into the ODS, and existing data in the ODS is updated to reflect the current status of the operational

system. The updates to the ODS occur within a short period of time of the updates to the operational systems – typically less than twenty-four hours. Unlike an EDW, an ODS does not contain summarized or historical data.

An ODS can be thought of as an operational data warehouse since it is used for the day-to-day management of business operations. If organizations were to reengineer their operational systems and make them subject-oriented, the resulting design would be somewhat similar to an ODS. It could be argued that an ODS has more in common with operational systems than with a data warehousing system. Many warehousing projects involve an ODS, rather than an EDW. An example of an ODS is a customer information system that integrates account and client data from various operational databases into a single database showing the current status of all customer accounts, i.e., it allows the business user to see a client as a single customer, rather than a set of different accounts.

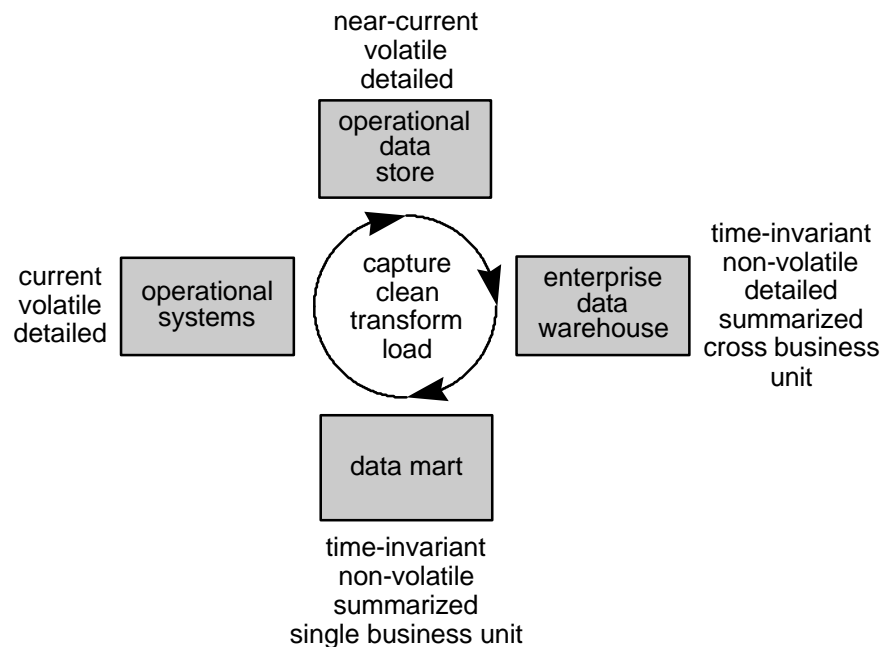


Figure 2. Types of data and data system

Moving from an ODS to an EDW

The ODS can act as a starting point for an EDW. An ODS contains clean, integrated, detailed operational data, and, therefore, the hardest part of any data warehousing project has been done, i.e., creating quality and usable data. When building an EDW from an ODS, the designer can focus on the issues of managing historical data and creating summarized information, rather than having to worry about data quality.

A data mart addresses a specific business problem

A **data mart** contains a subset of corporate data that is of value to a specific business unit, department, or set of users. This subset consists of historical, summarized, and possibly detailed data captured from operational systems, or from an EDW. Like an EDW, a data mart is used for short- and long-term business planning and decision making. Unlike an EDW, a data mart does not provide the capability to analyze data across multiple business units of the organization. It is important to realize that a data mart is defined by the functional scope of its users,

and not by the size of the data mart database. Most data marts today involve less than 100 GB of data; some are larger, however, and it is expected that as data mart usage increases they will rapidly increase in size.

The data in a data mart may be captured from one or more operational systems, or from an EDW. Data marts built directly from operational systems can often be developed in a matter of months at a significantly lower cost than an EDW. In the long term, however, the cost of integrating a multitude of data marts into a corporate information system could outweigh the short-term savings of using data mart technology. The solution is to build an EDW in parallel with data marts, and then use the EDW to populate the data marts.

A DATA TRANSFORMATION MANAGEMENT SYSTEM

The requirement – a managed approach

Regardless of whether data flows from operational systems to a data warehousing system, or from older operational systems to reengineered or downsized ones, the requirement remains the same – a tool that can capture, clean, transform, and integrate data, while at the same time handle the volume and complexity of the number of disparate data sources and targets involved. Such a tool is used in conjunction with other software products and processes to form what we will call in this report a *Data Transformation Management System* (DTMS). The next section of this report looks at a DTMS in more detail, and reviews product requirements in this area.

A DATA TRANSFORMATION MANAGEMENT SYSTEM

THE CLEANUP AND TRANSFORMATION PROCESS

Multiple tools may be required

The main objective of a data transformation management system (DTMS) is to improve the quality of the data flowing from a source system into a target system. A DTMS captures source data, cleans and transforms it, and then loads it into the target system (see Figure 3). A transportation capability is required to copy data to or from the transformation component if the capture or load processing occurs on a different system from the transformation processing.

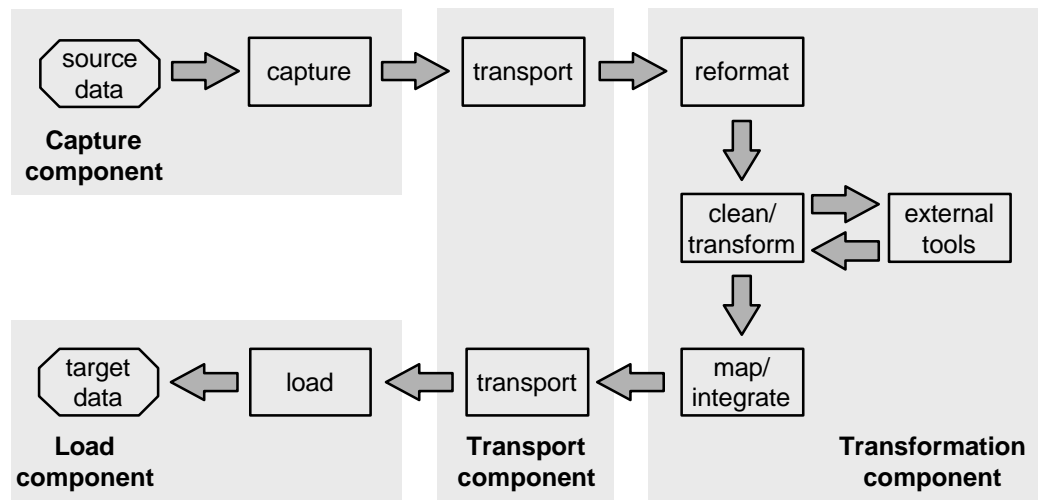


Figure 3. A data transformation management system

Building a DTMS

The main tasks involved in implementing each of the components of a DTMS are described below.

Capture

1. **Document the sources of data to be transformed.** This task involves not only identifying the databases and files containing the data of interest, but also analyzing and documenting the business meaning of the data, data relationships and business rules. This information may be documented in a variety of different places: on paper, in the programs that maintain the data, corporate repositories and data dictionaries, or in CASE tools. It is likely that there will be multiple sources of data with similar information, and it is important to identify the data source that is most likely to contain accurate and up-to-date information. If the source data is poorly documented, automated *rule-discovery* products can help in this process.
2. **Determine the products to be used for data capture.** Data capture programs and tools either extract all (or a subset of) the data in the source system, or capture the changes made to the source data by operational systems as they occur. In the former case, the capture process will typically use either an unload

utility or data manipulation language statements to extract the required source data. In the latter case, a recovery log or a database trigger is used to capture the changes into an intermediate file or database for processing by the data transformation subsystem. The method chosen by an organization will depend largely on the size of the data source and the amount of data to be captured. With an ODS, for example, it is usually impractical to reload the complete target database each time, and change data capture methods are often used instead.

Many data transformation products (see Step 6) have a capture component that directly accesses the source system using data manipulation language statements to extract the required data. Any given product, however, cannot possibly support all of the hundreds of database and file formats that exist in operational systems. As a result, additional data capture routines may have to be employed in some situations to create flat files that can be processed by a transformation tool.

3. **Check the integrity of the source data to verify that it conforms to the business rules and relationships identified in Step 1.** The objective here is to determine the quality of the source data. *Data analysis and auditing* products can be used to automate this task.
4. **Check the accuracy of the source data.** The objective again is to verify that the quality of the source data, but in this case the data values are checked to confirm that they reflect the real world. Often this can be done by sampling the source data, rather than auditing the whole database.
5. **Identify the tasks and products required for data cleanup.** If the analysis from Steps 3 or 4 indicates that data needs to be cleansed, then appropriate procedures and products need to be identified to do the work. Additional data cleansing may also be required to resolve issues concerning missing field values, the handling of freeform data such as name and address information, and so forth. A number of products exist to assist in this process. For example, companies like Postalsoft and Group 1 Software market products that clean up name and address data. This work can be done before the data transformation task or in conjunction with the tool used for doing data transformation

Transformation

6. **Document the rules and identify the products required for transforming mapping, and integrating the data into the format required by the target system.** Data transformation and integration involves the reformatting of the source data including files, records and fields, and the removal of data that is not required in the target system. It may also involve decoding and translating field values, adding a time attribute (if one is not present in the source data) to reflect the currency of data, data summarization, and the calculation of derived values.

There are many products on the market for doing data transformation. Some focus solely on transformation and integration, while others handle several of the tasks shown in Figure 3. Transformation approaches and products are discussed in the section “Vendor Approaches.”

Load

7. **Identify the products and techniques to be used for loading the data into the target system.** Data can be loaded into the target system using a load utility or data manipulation language statements. If a large amount of data is to be loaded, a load utility will normally provide better performance.

Transportation

8. **Evaluate the need for data compression and encryption if captured or transformed data is to be transported across a network.** For applications involving large amounts of data, compression techniques may aid transportation performance. Data encryption may be required for transporting highly sensitive data.

MANAGING A DTMS

“M” stands for management

One important aspect of a DTMS are the services provided for managing the development and runtime environments. Development involves primarily the creation of metadata that documents information about data sources and targets, and the business rules to be used for data transformation. This metadata should be stored in a DBMS-based repository or information directory. Requirements for this metadata store include a metadata import and export capability, an open and documented API, a documented and extensible metamodel, metadata versioning, and a reporting facility. Management of the runtime environment requires a GUI-based tool that supports configuration management, security and auditing, monitoring and tuning, transformation scheduling, and error recovery.

MANAGING COMPLEXITY

We have already discussed how the constantly evolving and changing business climate is leading to more complex data and application systems. For a DTMS, this translates into added complexity in the number of data sources and interfaces involved, and the amount of transformation and cleanup that has to be done.

Complexity varies by data system

As the illustration in Figure 4 shows, the level of complexity to be dealt with by a DTMS varies depending on the type of data system being managed. Moving data between operational systems involves a high level of data transformation and a large number of interfaces. An operational data store and an enterprise data warehouse also have a high data transformation requirement, but involve fewer interfaces. Data marts, on the other hand, assuming they are built directly from an enterprise data warehouse, have few interfaces and a lower level of data transformation.

Transformation power and interface management

The transformation power and interface management provided by a DTMS directly effects data quality and the ability of the product to scale to support the needs of a corporate information system.

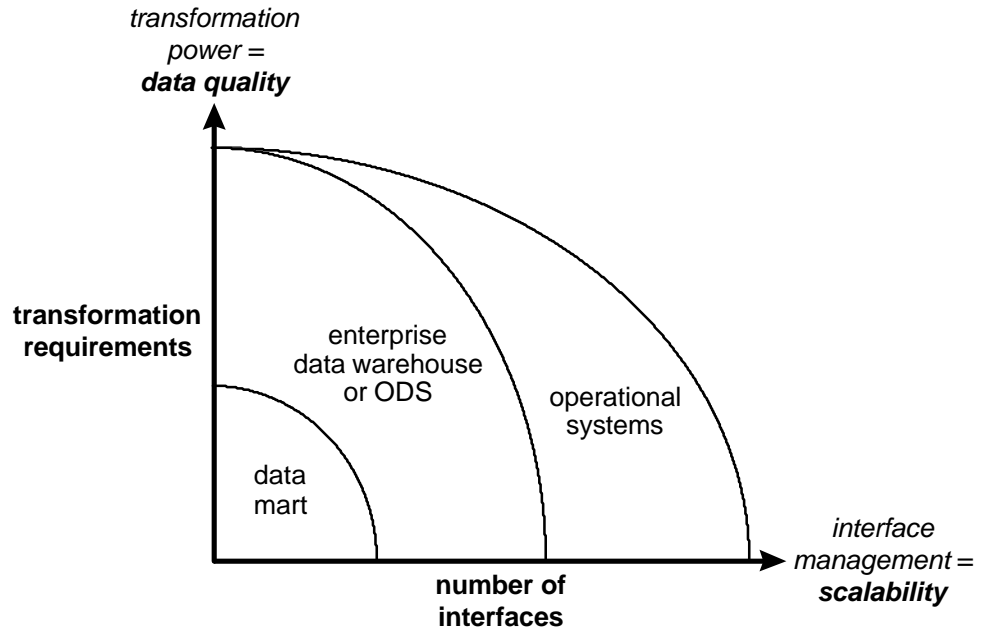


Figure 4. Data system complexity in a corporate information system

VENDOR APPROACHES

Integrated solutions

A DTMS consists of a number of components for capturing data from a source data system, cleaning and transforming it, and then loading the results into a target data system. These tasks can be carried out either by separate products, or by a single integrated solution. This section of the report focuses on the latter type, i.e., integrated products that support data capture, transformation and load. These products typically fall into one of the categories documented below.

Interface management is an issue for code generators

Code generators create tailored 3GL/4GL transformation programs based on source and target data definitions, and data transformation and enhancement rules defined by the developer. This approach reduces the need for an organization to write its own data capture, transformation and load programs. Code generation products employ data manipulation language statements to capture a subset of the data from the source system. Some also support the capture of changes to source data by processing the recovery log files of the source system. With most products, user-written programs or exits can be called for performing additional data transformation and enhancement.

Code generation products are used for data conversion projects, and for building an EDW, when there is a significant amount of data transformation to be done involving a variety of different flat file, non-relational and relational data sources. The main issue with this approach is the management of the large number of programs required to support a complex corporate information system. Each group of related data sources will result in a generated program that copies data from the source system to a target system. Often dozens, and possibly hundreds, of these *point-to-point* programs may be required to support the needs of an installation. Managing and coordinating such an environment is difficult and error-prone.

Vendors recognize this issue, and some are developing management components employing techniques such as work-flow methods, and automated scheduling systems.

Data replication has limited transformation power

Database **data replication tools** employ database triggers or a recovery log to capture *changes* to a single data source on one system and apply the changes to a copy of the source data located on a different system. Most replication products do not support the capture of changes to non-relational files and databases, and often do not provide facilities for significant data transformation and enhancement. These point-to-point tools are used for disaster recovery, and to build an ODS, EDW or data mart when the number of data sources involved is small, and a limited amount of data transformation and enhancement is required.

Data mart builders do not scale to an EDW or ODS

Rule-driven **data mart builders** capture data from a source system at user-defined intervals, transform the data, and then send and load the results into a target data mart. To date most products have supported only relational data sources, but products are now emerging that handle non-relational source files and databases. Data to be captured from the source system is usually defined using query language statements, and data transformation and enhancement is done based on a script or function logic defined to the tool. Some products also allow user-written code to be called for doing additional data transformation and enhancement. With most data mart builders, data flows from source systems to target systems through one or more

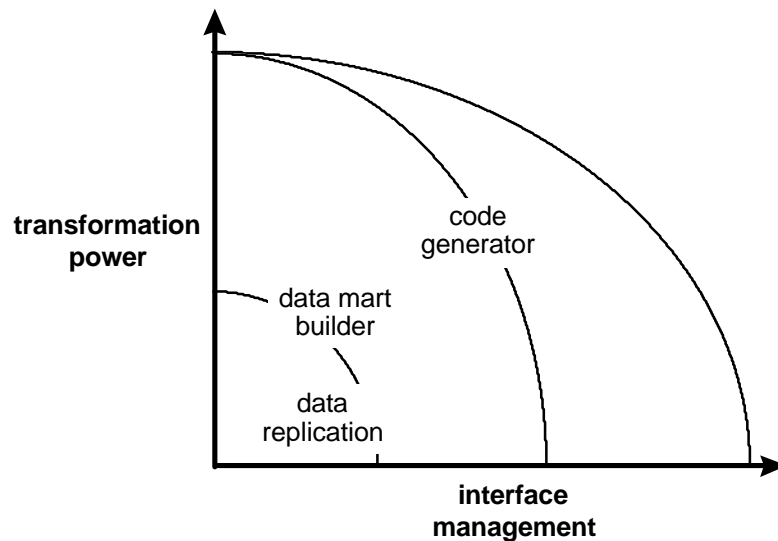


Figure 5. Data transformation product positioning

servers, which perform the data transformation and enhancement. These transformation servers can usually be controlled from a single location, making the job of managing such an environment much easier. At present performance constraints often prevent these products from being used to build a large ODS or EDW, and they are used mostly to build data marts (hence the term *data mart builder*, rather than *data warehouse builder*).

No vendor has a complete solution

Figure 5 shows how these products fit into the complexity graph introduced in Figure 4. As can be seen from the figure none of the approaches provides a single solution for building and managing a corporate information system involving a significant amount of data transformation, and a large number of data sources and targets. To address this problem, Constellar Corporation has developed its Constellar Hub system. Constellar expands on the transformation server concept in data mart builders by improving both transformation power and scalability in terms of the number of sources and targets that can be managed. The Constellar Hub is reviewed in the next section.

THE CONSTELLAR HUB

OVERVIEW

Designed for scalability

The Constellar Hub consists of a set of components supporting the Distributed Transformation Management capability outlined in this paper. The product is designed to handle the movement and transformation of large volumes of data for both data migration and data distribution in an operational system, and for capturing operational data for loading into a data warehouse.

Oracle-based hub and spoke architecture

Constellar employs a scalable *hub and spoke* architecture to manage the flow of data between source and target systems (see Figure 6). At the core of this environment are one or more multithreaded *transformation hubs* that perform data transformation based on rules defined and developed using the *Migration Manager* (refer to the section “Constellar Development” for more details). Transformation hubs run as servers on either UNIX or Windows NT, and make extensive use of Oracle tools, databases, and recovery features for handling and maintaining both data and metadata.

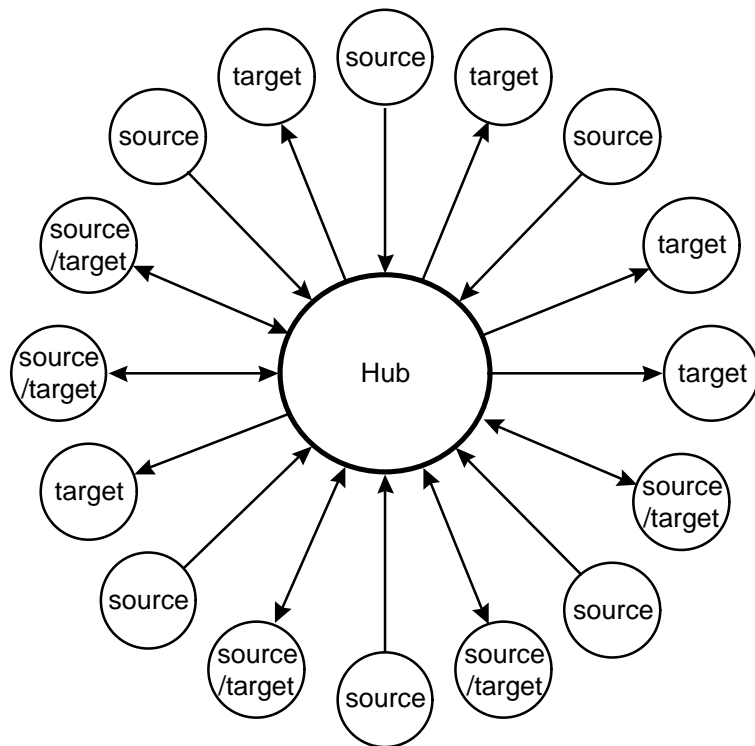


Figure 6. Constellar hub and spoke architecture

Each of the spokes shown in Figure 6 represents a data path between a transformation hub, and a data source or target. A hub and its associated sources and targets can be installed on the same machine, or may run on separate networked

computers. Product pricing is based on the number of hubs and spokes in the configuration. Constellar supports spokes (i.e., data sources and targets) for flat files, Oracle databases, and databases and files supported by Oracle’s database gateway products.

CONSTELLAR OPERATION

Provides the key components of a DTMS

The main components of the Constellar Hub are shown in Figure 7.

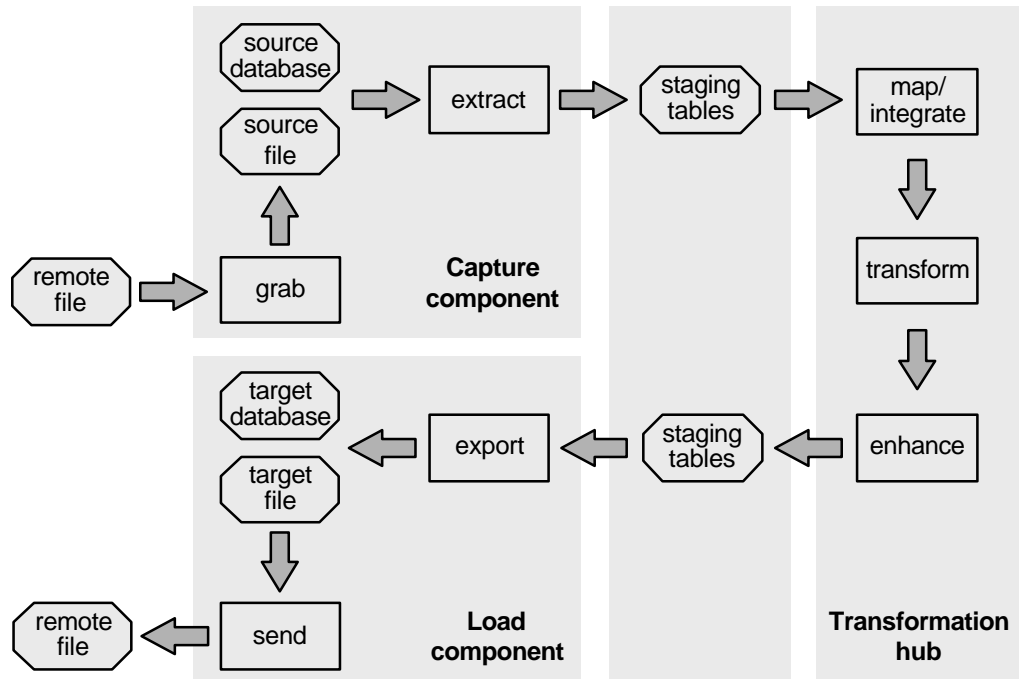


Figure 7. Constellar Hub components

Extract and **grab** constitute the Constellar data capture component. Extract selects records from local or remote databases using SQL data manipulation statements, and loads the records into internal staging tables for processing by the transformation hub. Records can also be extracted from local files, and from remote files (containing data unloaded from a remote database, for example) transferred to the local environment using grab. Extract can handle files containing repeating groups, fixed and variable length records, and multiple record types. Only the records and fields required for transformation are loaded into the staging area. Depending on performance requirements, the developer can choose to use the Oracle load utility or SQL to populate the staging tables. For simple file-to-file transformations, the staging area can be bypassed entirely, and selected records passed straight to the transformation hub. Multiple extract processes can be run on the same machine as the transformation hub, or a remote one.

The use of staging tables adds overhead to the transformation process, but has the advantage that when data needs to be acquired and integrated from many different source systems, and possibly at different times of the day, it can first be accumulated

into the staging area and then transformed when all the required data has been assembled. This makes it easier to manage a large transformation project.

The **transformation hub** performs the DTMS tasks of data cleanup and transformation. The hub supports:

- Record reformatting and restructuring.
- Field level data transformation, validation, and table lookup.
- File and multi-file set-level data transformation and validation.
- The creation of intermediate results for further *down stream* transformation by the hub.

Data is transformed in a two-step process. The *collate* step filters and joins source records, which are then passed to the *transform* step, where field-level transformation rules are applied. Output records from these two steps are inserted into the staging area. As with the extract process, for simple file-to-file transformations, the staging area can be bypassed, and records inserted straight into the target database. On multiprocessor machines, the work for each transform step can be broken down into several tasks running in parallel to increase performance.

Export and **send** represent the Constellar load component. Export writes transformed records from the staging tables to target databases and files, while send transfers result files to remote machines. Multiple export processes can be run on the same machine as the transformation hub, or a remote one.

CONSTELLAR DEVELOPMENT

Windows-based development system

The rules and procedures that control how data is extracted, transformed and exported as it flows from a source to a target system are defined using the *Migration Manager*. This tool supplies the developer with a Windows-based point-and-click interface for defining sources, targets and transformations. These definitions are stored in a set of relational tables known as the *metadata dictionary*. This dictionary can be shared by one or more transformation hubs, and can be replicated for recovery purposes using either Oracle symmetric replication, or Constellar Hub transformation services.

There are four main tasks involved in defining the processing to be performed to the Migration Manager.

1. **Specify the names and formats of the source and target databases, entities and attributes.** A database maps to a file or relational database, an entity to a relational table or record of a file, and an attribute to the column of a table or the field of a record. Metadata can be imported from Oracle table and column definitions, Oracle CASE, and from other third-party products using Oracle CASE Exchange. A COBOL copybook reader is also provided.

2. **If required, define *domain constraints to be associated with the attributes of an entity*.** These constraints are enforced during transformation to ensure that source and target data conform to a set of valid values.

3. **Define the *transactions and associated mappings for creating the target entities*.** A transaction defines one or more output entities to be created and the processing (the *logical unit-of-work*) required to create those entities. A transaction can capture and transform data from a single source database and load it into a single target database. A target database can be used as a source database for other transactions. The relationship between the source and target databases is defined in a *database mapping*. The entities and attributes in the source database to be used by a transaction to populate each output entity are defined by entity and attribute mappings. For operational purposes, the set of transactions associated with any given database mapping is known as a *migration* (see Figure 8). The developer can control the order in which transactions are run in a migration.

4. **Define and compile the *business rules that describe what processing is to be performed at each stage of transaction execution*.** Business rules specify, for example, pre- and post-transaction logic, flow logic that controls the filtering and joining of records, and attribute logic that defines how each target attribute is to be created or derived. Business rules are defined using the Transformation Definition Language (TDL), which is a high-level procedural language from which Constellar generates Oracle PL/SQL statements and C code. One very useful feature of TDL is the ability to interactively test individual business rules by typing in sample test data and running the rule to check for correct results. Developers can also embed PL/SQL procedures in TDL code, and call external PL/SQL and C routines to perform more complex business rules.

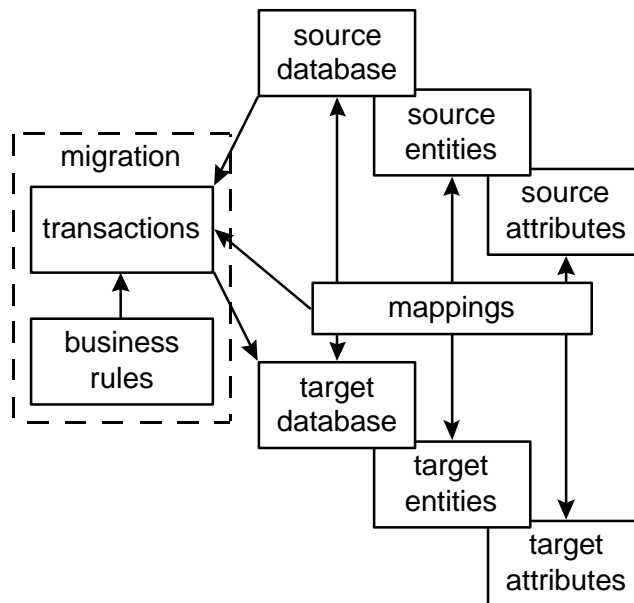


Figure 8. Migration definitions

MANAGING CONSTELLAR OPERATIONS

Wide range of management facilities

Administrators manage the Constellar runtime system using the *Migration Server*, which runs as a client application under Microsoft Windows. The graphical interface allows the administrator to configure and monitor each *migration* (a database mapping and its related set of transactions). It is not possible in this report to cover the wide range of options available to an administrator, but documented below is a sample of some of the main configuration and monitoring features of the Migration Server.

- The administrator can control which migrations are to be started, and the processes (grab, extract, transform, export, send) for each migration that are to run, or not run. Individual transactions of a migration can also be enabled or disabled. For each running process, the administrator can specify the type and level of tracing and debugging to be employed. A sophisticated scheduling facility is provided for automating the running of migrations. The scheduler allows the administrator to define business rules, which control, based on data content, the routing of transformed data to target databases. These rules can also be used to define interdependencies between migration jobs.
- The status of each migration – if it is waiting to start, is currently running, or has completed – can be displayed. There are also display options for showing record processing rates, and the number of source and target records processed so far by a migration. Statistics gathered during the running of a migration are stored in the metadata dictionary. These statistics can be used for auditing purposes and to create a historical log of migration operations.
- The trace feature can be used to track the progress of transaction execution within a migration – a snapshot of the current source record being processed can be displayed, for example.
- Error thresholds and conditions can be defined that cause processing to be stopped. Abnormal situations can trigger alerts, which can be handled by user-tailored PL/SQL procedures.
- During transaction processing any source records found to be in error, or that do not satisfy the conditions defined in business rules, are written to an error log. Administrators can review this log, fix any errors using a supplied editor, and resubmit the corrected records for processing.
- For multiprocessor machines, the number of parallel processes, and the number of source records to be handled concurrently by each processor can be specified.
- A reports package is provided for listing and formatting the contents of the metadata dictionary. Reports can be products showing the definitions created by the Migration Manager, and the configuration and runtime data created by the Migration Server. The metadata dictionary tables are fully documented and can be accessed by user-written applications.

A Data Transformation Management System

- Change management is handled using a version control option, which allows developers to check-in and check-out business rules and procedures from the metadata dictionary.

The above is just a partial list of the management services provided by the product. The important thing to note is that the Migration Server provides the management services required to support a large scale transformation hub environment.

SUMMARY

New technologies and products for reengineering and downsizing operational applications, building data warehouse systems, and developing web-based information systems provide an organization with a tremendous opportunity to modernize and grow its IT systems to reduce costs, increase profits, and to compete more effectively. The development and growth of these new systems must, however, be managed carefully if the full potential of these exciting new technologies and products is to be realized. Without a proper framework for building and integrating new IT systems, and reengineering old ones, the result will be a set of disparate and non-integrated systems that are difficult to manage and maintain, and that are unlikely to provide a good return on investment. A corporate information system provides such a framework.

A corporate information system involves a wide range of application systems supporting many different aspects of the business. Some applications support day-to-day operational processing, while others provide executives and managers with the business information they need for both tactical and strategic corporate decision making. These applications are deployed in a heterogeneous operating environment involving many machines networked together across both wide and local area networks. Such a network requires an efficient and scaleable Data Transformation Management System (DTMS) for distributing and transforming data between the various applications in the corporate information system. This paper has reviewed one such DTMS, the Constellar Hub from Constellar Corporation.

Constellar's DTMS architecture consists of an Oracle-based scaleable *transformation hub* that sends and receives data through a number of *spokes* that interconnect and integrate the many data systems that constitute the corporate information system. Constellar calls this interconnected environment an *application network*. As the network increases in size, hubs and spokes are added as required to handle the increased data handling and transformation requirements.

Besides scalability, two other key DTMS requirements for supporting large scale systems are transformation power and robust management facilities. Constellar's transformation hub supplies a powerful high-level transformation language known as TDL, which can be supplemented by user-written Oracle PL/SQL procedures and C code for performing more complex application-specific transformations. The product also includes a wide range of management services for automating transformation operations, system recovery, detecting and correcting errors, and monitoring and tuning.

Scaleable, and manageable DTMS capabilities like those provided with the Constellar Hub will become a key element in the future for enabling organizations to successfully build and deploy corporate information systems that supply quality data in support of business operations and corporate decision making.

ABOUT DATABASE ASSOCIATES INTERNATIONAL, INC.

Database Associates International is a consulting and training company specializing in leading-edge technologies in the fields of database, distributed computing, data warehousing, and Web technology.

ABOUT INFOIT, INC.

InfoIT is an information service from DataBase Associates International providing in-depth industry analysis about all aspects of new and evolving information technologies. InfoIT delivers this information to its clients through its bimonthly InfoDB magazine, product and technology reports, CD-ROM, and via its web server.



DataBase Associates International, Inc.

InfoIT, Inc.

Post Office Box 310

Morgan Hill, CA 95038-0310

408-779-0436 (voice)

408-779-3274 (fax)

<http://www.dbaint.com>

<http://www.infoit.com>

dbaint@dbaint.com (e-mail)

Copyright © 1997 by InfoIT, Inc.

Version 2. All rights reserved.

About DataMirror Corporation

DataMirror (Nasdaq: DMCX; TSE: DMC) delivers solutions that let customers integrate their data across their enterprises. DataMirror's comprehensive family of products includes advanced real-time capture, transform and flow (CTF) technology that gives customers the instant access, integration and availability they demand today across all computers in their business.

Over 1,400 companies use DataMirror to integrate their data. Real-time data drives all business. DataMirror is headquartered in Toronto, Canada, and has offices worldwide. DataMirror has been ranked in the Deloitte and Touche Fast 500 as one of the fastest growing technology companies in North America.

DataMirror[®]

www.datamirror.com

1 800 362-5955