

Issues and approaches for migration/cohabitation between legacy and new systems

Rodolphe Nassif
U S WEST Advanced Technologies
4001 Discovery Drive
Boulder, CO 80303

Don Mitchusson
U S WEST Communications
930 15th Street
Denver, CO 80202

Abstract: Corporate Subject Data Bases (CSDB) are being introduced to reduce data redundancy, maintain the integrity of the data, provide a uniform data access interface, and have data readily available to make business decisions. During the transition phase, there is a need to maintain Legacy Systems (LS), CSDB, and to synchronize between them. Choosing the right granularity for migration of data and functionality is essential to the success of the migration strategy. Technologies being used to support the transition to CSDB include relational systems supporting stored procedures, remote procedures, expert systems, object-oriented approach, reengineering tools, and data transition tools. For our Customer CSDB to be deployed in 1993, cleanup of data occurs during initial load of the CSDB. Nightly updates are needed during the transition phase to account for operations executed through LS. There is a lack of an integrated set of tools to help in the transition phase.

1.0 Introduction

Corporate data is currently controlled and accessed through legacy systems (LS). LS were developed independently. No mechanisms were in place to ensure the consistency of data replicated in different LS. Business rules are buried into each LS code. Different business rules may be applied in two LS to the same piece of information. All of the above resulted in having corrupted and replicated data. For example, at U S WEST, there are tens of copies of billing, customer and product information. It was estimated that maintaining correct information on customer addresses (billing and service) could save tens of millions of dollars. Developing new applications is hindered by the need to use LS interfaces that are mostly based on terminal emulation or on transaction files. The cost of maintaining LS code is also very prohibitive. In order to partially address these problems, we have adopted a strategy of building relational corporate subject databases that would maintain the consistency of data and provide a programmatic interface to applications. A phased approach is taken to populate the CSDB and to migrate functionality from LS. Section 2

gives a partial view of the Customer CSDB to be deployed in 1993. Section 3 covers the approaches we are taking to support the transition effort for new applications.

2.0 Customer CSDB

The customer CSDB maintain information about U S WEST Customers. A Customer may be an individual or an organization. Customers may have parent organizations. Businesses have standard industry codes. Billing names for business customers may include an individual name as well as a business name (e.g. John Doe Doing Business As Terminators). Customers also may have listed names. Multiple different addresses may be associated with a customer, e.g. billing, listed, and service. Customers are associated with market units according to some business rules. For example, customers with less than a certain number of exchange lines at their biggest location are associated with a specific market unit (MU). Customers are usually associated with one market unit. However, there is a list of customers (e.g. franchises) that can be shared by different market units. In some cases, customers should remain associated with the same MU for a given period of time before being switched to another MU.

The Customer CSDB will contain data for which it is the source of record (SOR) as well as data whose SOR is in LS. For example, customer demographic data will appear only in the customer CSDB, which is therefore the SOR for this data. For data duplicated in the Customer CSDB whose SOR is in LS, it is necessary to synchronize the CSDB with data in the LS and vice versa. In order to ensure the correctness of CSDB, it is necessary to perform some verifications at the front end of the service negotiation process versus the current state where data cleanup is done at the back end. Currently, service order representatives, using existing LS, take customer service orders and assign a market unit code and a standard industry code based on the information received from the customer. The service order is submitted to a service order processing system that interacts with the network. Once the service order has been com-

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing

Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

SIGMOD /5/93/Washington, DC, USA

© 1993 ACM 0-89791-592-5/93/0005/0471...\$1.50

pleted, it is submitted to Customer service Record Information System (CRIS). CRIS generates a daily log of its activities. Extracts from CRIS are used to populate a database where customer to customer account assignment process is done. This process is based on matching of names and addresses on new service orders to existing accounts. Errors might go undetected at this stage (e.g. a company name may be misspelled and therefore interpreted as a new customer). Some errors detected at this stage would be difficult to correct. For example, an address might be ambiguous (e.g. the directional field is not specified), or non existing (e.g. misspelled street name, wrong city or zip code). Discrepancies might also occur. The automatic system might assign a customer to a different market unit than the one specified on the service order by the service rep. New applications attempt to detect errors as early as possible. For example, addresses on service orders would be checked while the customer is on line and before the service order is submitted for processing. The customer to customer account assignment process would be done on line.

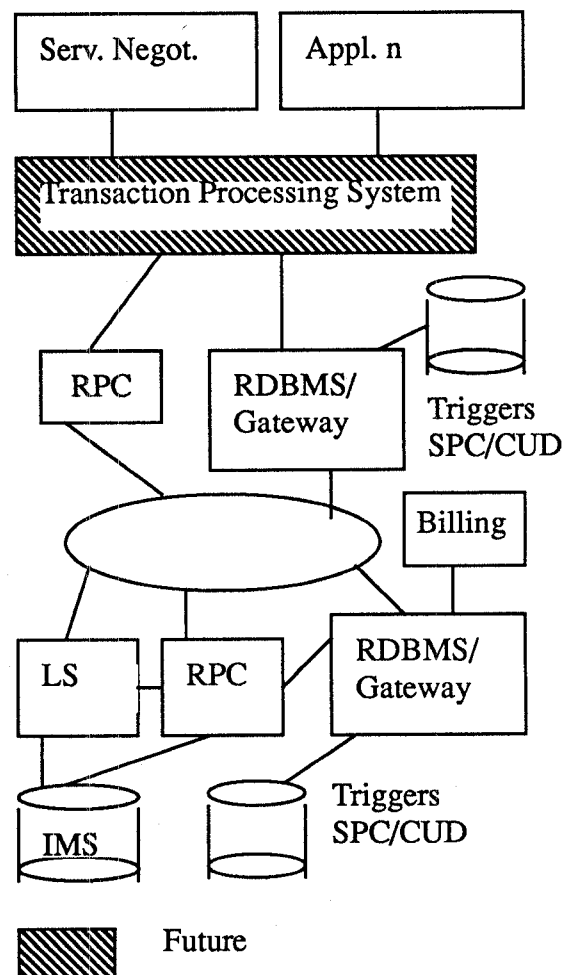
3.0 Approach for data migration/co-habitation

The approach taken by U S WEST is to create Corporate Subject Data Bases in an incremental way. The Corporate Data Model has been specified using an information engineering tool and is continually refined. The Customer CSDB will be deployed in fourth quarter 1993. Several issues are being addressed: Maintaining consistency of CSDB, migration of functionality, transaction processing, heterogeneous data access. Object-oriented and re-engineering technologies are being investigated to provide possible solutions.

3.1 Maintaining consistency of CSDB

There are several levels to enforce business rules. At the lowest level DBMS primitives are used. Referential constraints fall into this category. At the second level triggers are written to ensure that a business rule is satisfied, e.g., checking that a customer orders all of the required components for a product, or that a customer does not order incompatible products. However, due to the limitations of RDBMS in the area of the language features supported to define triggers and stored procedures (e.g. inability to call external programs and get information back), a third level is needed. Constraints are enforced at this level through procedures or through a rule based system. In order to match names and addresses external tools need to be invoked. Business rules based approach might be the right way to specify some constraints versus a procedural approach. Create, Update, and Delete procedures (CUD) are used to control updates to CSDB. Most CUD procedures are implemented as Stored Procedures or associated to

packages. Regular procedures are used when external access is needed. These procedures need to be invoked through remote procedure calls. For customer to customer account assignment, a procedure to check for a match of a customer name and address to existing customers/locations would be defined. It would invoke the rule based system and procedural software to check for matches on customer name/address. Depending on match results, either a new customer and/or site is created, or the service is associated with an existing site for the customer.



3.2 Migrating Functionality

The approach to migration of LS consists of taking a vertical slice of a LS and duplicating its functionality in a new system. As deployment of a new system is done in phases, and because LS interact with other LS, there is a need to forward operations from the new systems to LS, and to propagate operations from LS to new systems. At U S WEST, most of our LS provide logs of all the operations submitted to them. These logs are being

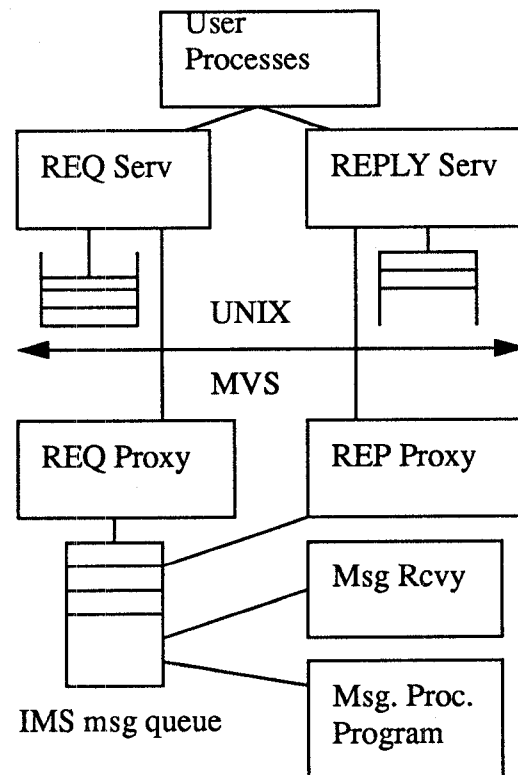
used to propagate changes to new systems in a batch mode. The first two subject databases to be deployed will be customer and product databases. New applications will be interacting with these databases. When all the information needed by a new application, e.g. billing, is available in the CSDB, then corresponding LS functionality would not be needed anymore except to support other LS. During the transition, it is very common to have: a LS that executes some functionality on its own data; a CSDB that partially replicates the data contained in the LS; a new application that works with CSDB and forwards operations to LS; the new application is deployed initially in a limited number of sites; and it partially replicates some of the functionality of the LS. Data replicated in the CSDB might be more accurate due to the additional processing provided by new systems or by the CUD transactions. New systems would have a batch component, to run nightly for example, to handle operations executed on the LS that are replicated in the new system, for the data stored in the CSDB, through the sites that are not part of the current deployment for the new system. Furthermore, a true-up process is needed, monthly for example, between the CSDB and the LS databases. The true-up could account for updates done on LS data through some functionality that has not yet been replicated in new systems. A true-up could also propagate updates to legacy systems from the CSDB. The effort needed in the transition phase is often underestimated. Handling the transition is very critical to the success of the deployment of new applications.

3.3 Transaction Processing

Current relational database vendors provide 2-phase commitment. Stored Procedures (SP) are implemented without a Begin/End transaction so that a SP can call another SP, and an application can invoke these SP from within a transaction. Invoking a remote procedure from within a transaction introduces an additional complexity as the remote procedure would execute in a different process and environment than the process that started the transaction. An intermediate solution is to have the remote procedure start and commit its own transaction. Compensation procedures (undo) would be executed if the origination transaction is rolled back. For example, a customer might decide to cancel a service order after network facilities have been assigned to the customer. At the current stage we have two disjoint environments: transactions supported by relational vendors, and remote procedure call technology. We have used on some projects ONCTM and NCSTM rpc mechanisms; DCE will be used in the future. Transaction Processing Systems that support open systems interfaces are starting to appear in the market (EncinaTM, TopEndTM, TuxedoTM).

3.4 Heterogeneous data access

Although the bulk of CSDB will be in relational systems, some LS would still be the source for some categories of data. For example, the Work Force Administration/Dispatch Out (WFA/DO) LS uses IMS and might remain the source for technician job information. There is a need therefore for an environment that provides access to heterogeneous data. Some tools are starting to appear on the market that provide a conceptual relational view of information that may be stored in relational, network, hierarchical databases, or flat files. The databases might reside on UNIX, MVS or other operating systems. These tools are oriented more towards ad-hoc queries. They are not appropriate to handle invocation of existing procedures (e.g. message processing programs on IMS). Efficient handling of procedural code is needed for production systems where thousand of users may be making the same query. We developed a queued based message processing mechanism that ensures delivery of messages to a resource manager (i.e. IMS) and replies to application processes. We have used ONC remote procedure calls over IBM 3172 TCP/IP connection for communication from UNIX to MVS.



3.5 Object-Oriented approach

Object-Oriented approach has been used in the user interface area. However, it provides the right framework to support migration/co-habitation with new systems. An object would encapsulate the behavior of a LS component. Operations would be forwarded originally to the encapsulated LS. As some slices of the LS are migrated to the CSDB and to new applications, the implementation of specific procedures are changed to reflect the new implementation. As the interface to the objects did not change, then applications using these objects do not need to be affected. For some LS, e.g WFA/DO, new interfaces are being provided through a contract interface to provide application programming interfaces as alternatives to terminal emulation access. These contracts are at a high level of granularity. Some of these contracts would be later decomposed into lower level contracts. By using an OO approach, we reduce the changes needed for applications interacting with these contracts. For example, a bulk job download contract accesses WFA/DO data, and might invoke originally the interface of a WFA/Control (WFA/C) application to get some text documents. When WFA/C implements a contract interface to get the text documents, then the WFA/DO contract implementation would be changed to interact with the new WFA/C contract. However, a Field Access System software components that uses the WFA/DO contract would not be affected. Object-Oriented approach is also useful for distribution transparency.

3.6 Re-Engineering

Selective migration of slices of LS can be facilitated by re-engineering tools that assist in understanding existing LS code in order to abstract data, business rules, and processing.

For the data component, a couple of disintegrated tools are being used for producing the CSDB physical data structures. The Corporate Logical Data Model (LDM) is managed in IEF. From IEF the LDM is uploaded to the Corporate Repository, UIR, where LDM compliance with corporate standards is ensured. To perform physical data design and develop the Physical Data Model (PDM), the LDM is downloaded to another tool, the Bachman Data Design Tool. Upon its completion, the PDM is uploaded back to UIR from which DDL is generated compliant with corporate standards. This illustrates the need for an integrated development and transition tools.

4.0 Conclusion

Corporate Subject Data Bases are being implemented at U S WEST as a mean to provide a single source of information, data consistency, availability, and provide a programmatic interface to applications. New applications are slicing-off functionality from LS. Co-habita-

tion is handled currently on an add-hoc basis with proliferation of interfaces between applications. There is a need for a comprehensive platform to support such environments. Transaction Processing Systems would be an important component.

Acknowledgment: We would like to thank Jackie Anderson, Howard Gluckman, Elaine Hill, and Linda Schmillen for information on the CSDB. We would also like to thank Pankaj Goyal, Patrick Pfeffer, Bala Ramakrishnan, and Jianhua Zhu for their comments that enhanced the quality of this paper.