RDC Speaks Your Business Language

For over 35 years, RDC Datacentrum has established and developed information, communication, and technology (ICT) services for the Dutch Automotive sector. Originally, RDC was set up as a supplier of statistical information for the vehicle sector but then expanded into the computerization center of the vehicle branch by offering services such as motor vehicle testing, used car information exchange, and much more.

BUSINESS BACKGROUND

Every day in the Netherlands alone, over 100,000 transactions are processed by RDC via the Internet from dealers, manufacturers, importers, car companies, auto body repair shops and car leasing companies. Almost 170 specialists at RDC develop and support the full services of ICT solutions to the automotive sector, from network architecture to support desk, from application development to financial services.

SYSTEM PROFILE

Located in close proximity to the RAI, Amsterdam's famous Congress Centre, RDC Datacentrum's employees keep the myriad activities and services of a modern data processing environment up and running smoothly. IBM mainframes host the z/VSE operating system and the production data of RDC is stored in IDMS databases. AIX-based systems act as servers to host the various Client/Server applications that use DB2/UDB as their DBMS. The preferred programming language used on the mainframe is COBOL. Open-systems applications are developed in C, PHP and Java.

BUSINESS ISSUE

About 11 years ago, RDC decided to make their IDMS databases available to open systems applications. System Manager Tom Boeken describes the scenario, "A new client/server application was about to be developed and we needed direct access to our IDMS databases on the VSE mainframe. At the time, we made a decision to use a US software package

that permitted us to use SQL syntax to retrieve data from IDMS. We had to develop an interface module in C so that we could use the product within our AIX applications." The client/server a pplication went into production and other applications followed, all using the same in-house developed interface and middleware.

ICS Manager Richard van der Nat added, "In early 2004, we were surprised to learn that the software company whose package we had licensed – which had become very strategic to us – was sold and the new owner of the package decided to drop support of the VSE version. We had to make a decision on an alternative solution as soon as possible. That was when we learned about B.O.S. Software and tcACCESS."

TECHNOLOGY SOLUTION

tcACCESS is an innovative software solution that enables IBM mainframe data and applications to integrate with client/server, Web and SOA technologies – without the need for mainframe knowledge or know-how, or programming effort. The production use of tcACCESS started at the end of 2004 and has been in use every day since. Richard van der Nat was so pleased with the solution that he soon contacted B.O.S. again to learn more about their new tcVISION product to help in the migration from IDMS to relational databases.

According to van der Nat, "Our focus and concern was the real-time synchronization between IDMS and DB2/UDB and Microsoft SQL Server. The only – but major – problem we were facing at that
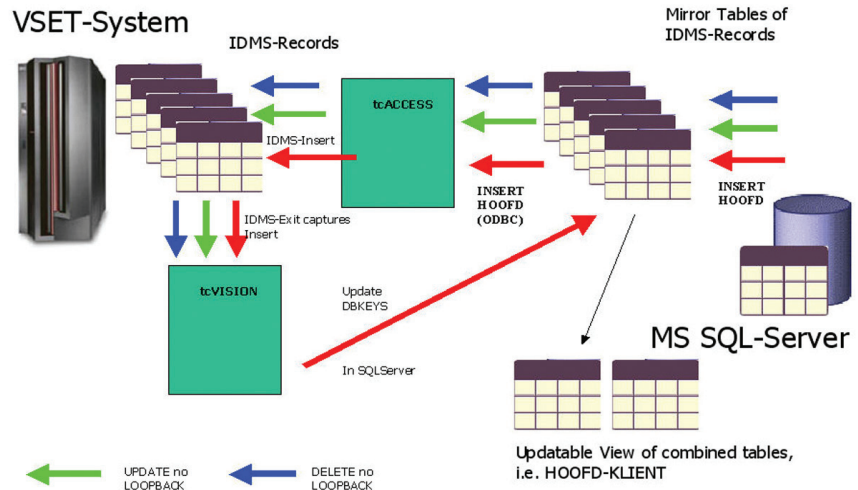
Treehouse S O F T W A R E

time was that neither a VSE version nor support for IDMS was available. We were confident in the technical capabilities of B.O.S. and so we soon ordered a VSE/IDMS version of tcVISION. tcVISION's job is to automatically capture all changes in IDMS and to synchronize these changes with DB2/UDB and SQL Server databases. The applications affected run on UNIX and Windows.

"The product was delivered to us in the promised timeframe and we were able to quickly begin the implementation process. tcVISION has been in use since summer of 2005."

Marcel Versteeg, responsible for the synchronization project, says: "tcVISION automatically captures all changes applied from our VSE applications to IDMS and propagates them to DB2/UDB on AIX and SQL Server on Windows. We have mirrored our IDMS networking architecture in the relational world. Changes performed by our Windows and UNIX applications

are captured by database triggers, and are passed as SQL statements to the VSE system. We use the tcACCESS ODBC driver for this. The tcACCESS VSE component takes the SQL statements and performs the corresponding IDMS updates. The challenge is to ensure that these changes are not being propagated back by tcVISION. IDMS is a network database and supports "N to N" relations. To keep both worlds in sync it is mandatory that so called "Foreign Keys" must be propagated back as UPDATES from IDMS, when a new record has been inserted into IDMS. This is IDMS-specific and rather complex. tcVISION masters this challenge without any difficulties."

Richard van der Nat: "We are glad that we are partnering with B.O.S. Our experience so far has been very encouraging. B.O.S. is a company that listens to what the customers say and then provides solutions. Taking tcVISION on board was -- and still is -- a good decision."

The diagram above shows the bidirectional synchronization between the VSE IDMS database and various Windows applications that store their data in SQL Server databases.

The IDMS-specific DBKEYS (internal database pointer) have been defined in MS SQL Server as "Foreign Keys". tcVISION captures all real-time changes performed by the VSE programs and propagates the changes to SQL Server. The DBKEY Foreign Keys are automatically maintained.

Changes performed to SQL Server by the Windows applications are captured using triggers and passed to VSE via the tcACCESS ODBC driver. The mainframe tcACCESS VSE SQL Engine performs the required IDMS changes.

tcVISION recognizes all changes performed by tcACCESS. UPDATEs and DELETEs are not propagated back to Windows.

In the case of an INSERT, the DBKEY in the SQL Server database is initially NULL. After the INSERT has been applied to IDMS, IDMS creates a new DBKEY for the record. tcVISION captures this DBKEY, rejects the INSERT and instead passes an UPDATE statement to SQL Server to apply the new DBKEY.

. . . .