Laplanda MES_SQL_Connect software

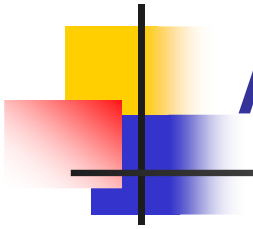
Manufacturing Execution System (MES) SQL (Structured Query Language) DB Connectivity

The Challenge

Every business is unique, however its organization obeys the strict rules of the market. The purpose of Laplanda Business Management software is to connect the business result to the real world by creating a standard procedure which will lead to clear connection and adoption in the market.

The Solution

- A Tier-3 architecture approach has been used to develop user interfaces and business services.
- The SQL queries was used to develop the data services with approach to minimize rework when new modules is integrated into the MES. Eventually, a stronger coupling between test sets, production control applications and the MES is developed.



Abstract

- Our MES application is tracking the fabrication of each component. All relevant data is recorded (Temperature, technician ID, work center ID, time, procedure, out-of-specs components, defects, etc…) and can be retrieved to analyze defects and improve quality. Reporting is done using Crystal Report.
- Purchase orders from customers are transformed into work orders and the production is tracked trough the MES.
- The products and their bill of material (BOM) are defined in the database, along with the work centers and the travelers. Change requests are also managed by the system.
- The solution is expendable, and the system may eventually include the Raw Material Treatment, Planning, Bidding, Invoicing, Procurement and Inventory Management modules.

Increasing productivity and enabling growth through a MES

- The project initiated from the need of Precision Design Machine Inc (PDM) to track Transfer Fab components and to replace its paper-based tracking processes.
- PDM wanted to keep and retrieve the fabrication history of every component and assembly product.
- PDM also wanted to set the corner stones of a production system that could grow with the company and increase the overall Fab productivity…

System features

- Basically, the system tracks the components from ordering to wrapping. Once a client order is taken, the information is transferred into the MES. Production orders and work orders are created using the appropriate traveler (which is selected in the product definition phase). On the shop floor, there is usually a computer station for every work center (which is associated with a given production operation). When a technician begins a given operation, he scans the bar code of the current component. Automatically, this triggers the system to obtain from the database the information related to activities to be performed for that component for the actual operation. This way, every work order is tracked, and production's progress is monitored constantly.
- Since products are often customized for a given Fab, the MES has been developed with flexibility in mind.
- Consequently, users can easily modify or define new products, operations (i.e. sequences of activities), travelers (i.e.sequencesof operations) and equipment in a timely matter. These actions are considered to be engineering changes and are tracked for later use. Request, consultation and approbation are performed using the MES.
- PDM Inc also required a high level of security in order to fulfill quality requirements and to pass ISO and BellCoreauditing. Authorizations are required and performed using electronic signatures. Access to modules is controlled and user groups are defined. The system manages access and modification privileges in order to refrain unauthorized users from seeing sensible data or from modifying products and/or orders.

Future enhancements

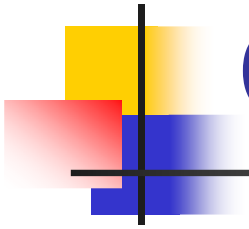
- The system has been designed with scalability and expendability in mind. PDM Inc is a fast-growing micro component manufacturer. Production is expected to increase exponentially over the next few years, and several Fabs will be opened. The new MES will therefore run in multiple Fabs and will be used by hundreds of users. Many additional modules will be added over the next year and integrated within the system. These modules support processes such as Raw Material Treatment, Planning, Bidding, Invoicing, Procurement and Inventory Management. Depending on our client requirements, custom or off-the-shelf software will be integrated to provide the best solution.

Development methodology

- The project started with a functional analysis that lasted several weeks. Then, system specifications were
- documented to provide development process with a source of information to refer to. After acceptance of this document by PDM Inc, it was created the global design for the system, outlining the various system modules and their interrelations. Every system module was then subject to a detailed design. During that time, SQL quires for the database were created and was defined the database procedures.
- From there, it was started translating detailed designs into source code that followed AKOR coding standard. Each unit and module tested accordingly. Upon success, each module was integrated in the next internal release of the system, which was then tested for performance and robustness.

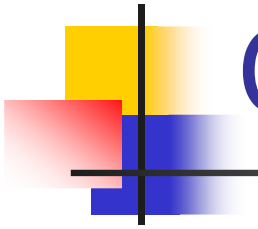
Technology and development tools

- The system has been developed using a tier-3 architecture. This approach was used as a good software engineering practice to minimize rework and risks. The representation language used for the architecture was UML and the tool was Microsoft Visual Developer.
- User Interfaces, Business Services and Data Services were coded in NI coding. Data services also made use of the SQL Toolkit.
- PDM Inc already selected Microsoft as its general platform supplier. The selected development and deployment platform was then Windows 7 and MS SQL Server 7.0.



OLE DB Providers

- The Microsoft Universal Data Access (UDA) platform allows applications to exchange relational or non-relational data across intranets or the Internet, essentially connecting any type of data with any type of application. OLE DB is the Microsoft system-level programming interface to diverse sources of data. The Microsoft ActiveX Data Object (ADO) standard is the application-level programming interface.
- The Microsoft Data Access Components (MDAC) are the practical implementation of the Microsoft UDA strategy. MDAC includes ODBC, OLE DB, and ADO components. MDAC also installs several data providers you can use to open a connection to a specific data source, such as an MS Access database. The OLE DB provider for ODBC acts as a conversion layer between OLE DB interfaces and ODBC.



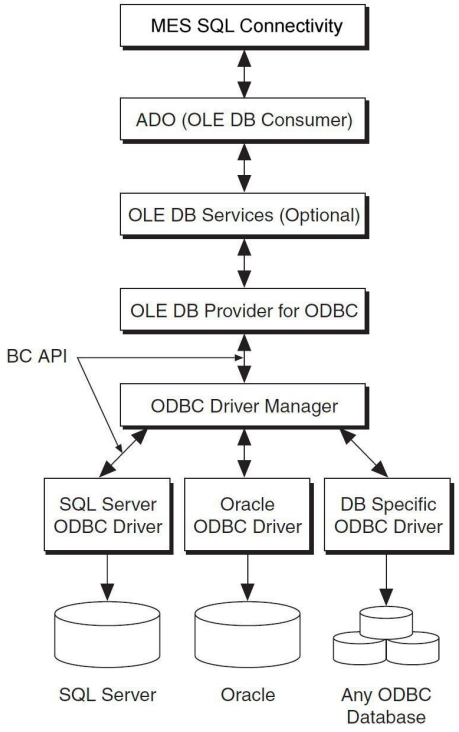
OLE DB Standard

OLE DB specifies a set of Microsoft Component Object Model (COM) interfaces that support various database management system services. These interfaces enable you to create software components that comprise the UDA platform. OLE DB is a C++ API that allows for lower-level database access from a C++ compiler. Three general types of COM components for OLE DB include:

- • **OLE DB Data Providers**—Data-source-specific software layers that access and expose data.
- • **OLE DB Consumers**—Data-centric applications, components, or tools that use data through OLE DB interfaces. Using networking terms, OLE DB consumers are the clients, and the OLE DB data provider is the server.
- • **OLE DB Service Providers**—Optional components that implement standard services to extend the functionality of data providers. Examples of these services include cursor engines, query processors, and data conversion engines.

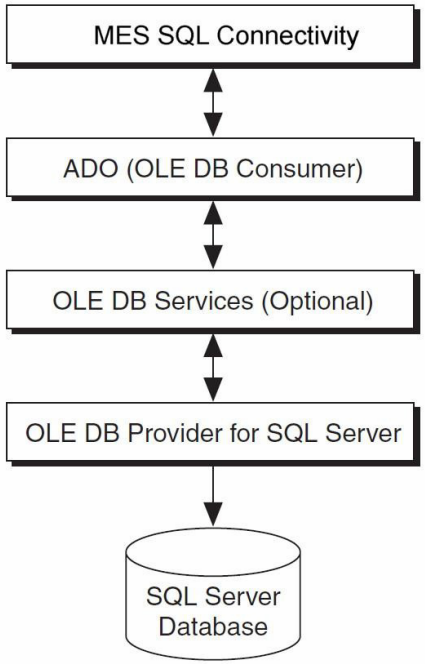
OLE DB Provider for ODBC

- The hierarchy of data interface layers between MES_SQL_Connectivityand a database using the OLE DB provider for ODBC appears in this Figure.
- MDAC 2.0 and later provide OLE DB providers for SQL Server, Jet, and Oracle database systems. Using native providers is much faster than using the OLE DB Provider for ODBC because native providers eliminate the need for both the OLE DB to ODBC conversion process and for the ODBC driver and ODBC Driver Manager layers. For this reason, always use the native OLE DB data provider for the data source you are accessing if a native provider is available.



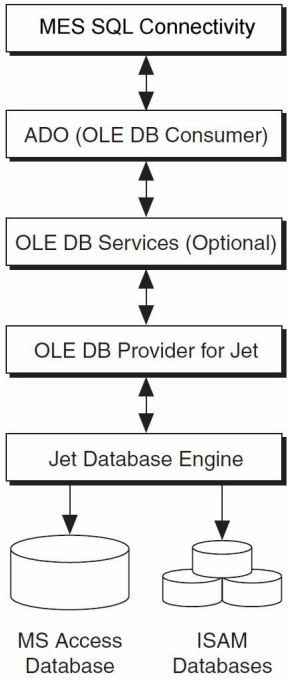
OLE DB Provider for SQL Server

- The OLE DB provider for SQL Server, shown in Figure 2-2, exposes data stored in Microsoft SQL Server 6.5 or later databases.
- This Fig. shows the Communication Path between MES_SQL_Connectivityand an SQL Server Database using the Native OLE DB Provider.



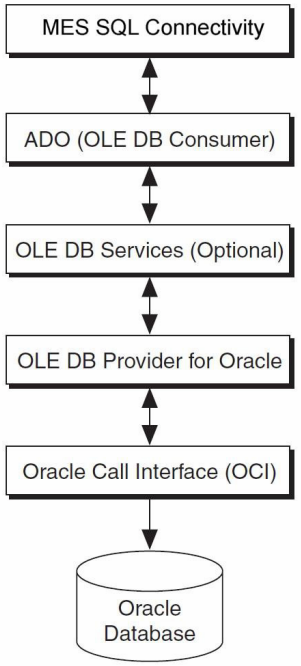
OLE DB Provider for Jet

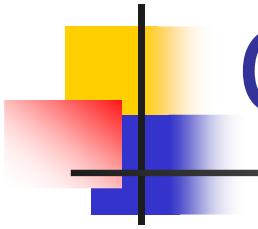
- The OLE DB Provider for Jet uses the Microsoft Jet database engine to expose data stored in Microsoft Access databases (.mdb) and numerous Indexed Sequential Access Method (ISAM) databases, including Paradox, dBase, Btrieve, Excel, and FoxPro. The Jet database engine is included with Microsoft Access and is the underlying Database Management System (DBMS) of Microsoft Access. Visual Basic for Applications is the
- host language for the Jet DBMS.
- Data Access Objects (DAO) is the Jet interface for using the Jet database engine programmatically. DAO is a COM component that provides custom applications with the power and flexibility of the Jet database engine in a simple object model. DAO is also language-independent. Any programming language or toolkit that supports OLE Automation can use DAO and the Jet database engine.
- Despite the availability of the OLE DB Provider for Jet and comparable benchmarks, some of the functionality of DAO, such as data definition and security, is not available in the OLE DB Provider for Jet.



OLE DB Provider for Oracle

- The OLE DB provider for Oracle exposes OLE DB interfaces for retrieving and manipulating data stored in Oracle 7.3.3 or later databases. The OLE DB provider for Oracle is implemented as a layer on top of the Oracle native API, the Oracle Call Interface (OCI). Refer to the Oracle Web site at www.oracle.com for more information about the OLE DB provider for Oracle.
- This Fig. shows the Communication Path between MES_SQL_Connectivityand an Oracle Database using the Native OLE DB Provider.





Custom OLE DB Providers

- An advantage of UDA is the ability to develop custom OLE DB data providers because UDA enables standardized access to data sources beyond Microsoft products and the popular relational database systems.
- If you need access to a data source that does not provide an OLE DB data provider and does not support ODBC, you can create custom OLE DB data providers that can expose any data source. For example, you can develop custom OLE DB data providers for data sources such as the following:

  - Personal address book
  - Windows registry
  - Scheduled tasks
  - Shared memory

Data Type Mapping

Database Connectivity Toolkit Data Types

MES_SQL_Connectivity	SQL Data Type	Description of SQL Data Type
string	CHAR (x), VARCHAR (x)	**CHAR**—Fixed character data such as CHAR (16). Extra is filled with spaces. **VARCHAR**—Varying character data. Does not pad with spaces.
long	INTEGER	Precision depends on the specific SQL implementation; database developer cannot specify the precision.
single	REAL	Single-precision floating-point number determined by the OS implementation of a SGL.
double	DOUBLE PRECISION	Double-precision floating-point number determined by the OS implementation of a DBL.
date/time	DATE, TIME (p)	**DATE**—Length of 10 positions in the form: YYYY-MM-DD. **VARCHAR**—Has the form: HH:MM:SS.SSS... specified by p.
binary	BINARY (n), VARBINARY (n)	**BINARY**—Fixed length binary string with maximum length n. **VARBINARY**—Variable length binary string with maximum length n.

Data Types

MES_SQL_Connectivity	Database Connectivity Toolkit Data Type
8-bit integers	long
16-bit integers	long
32-bit integers <= 2147483647	long
32-bit integers > 2147483647	string
8-bit enum	long
16-bit enum	long
32-bit enums <= 2147483647	long
32-bit enums > 2147483647	string
64-bit integers	string
64-bit enums	string
Single numeric	single

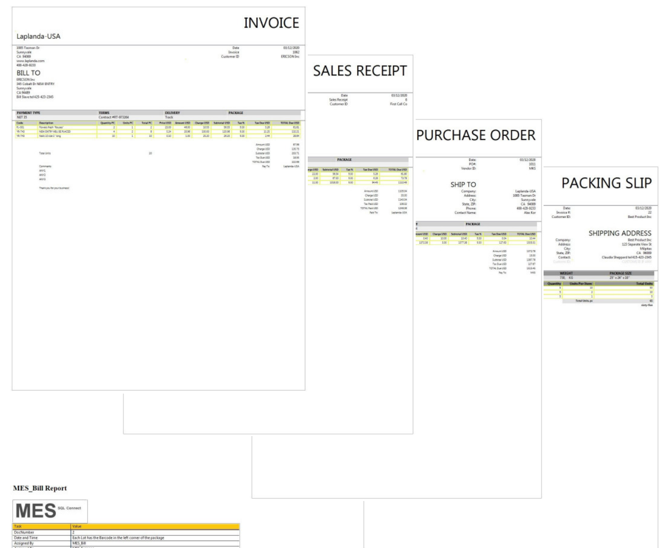
Data Type Mapping (continue)

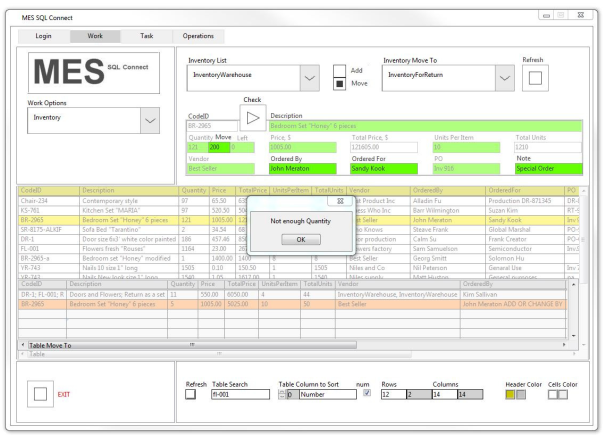
MES_SQL_Connectivity Data Types

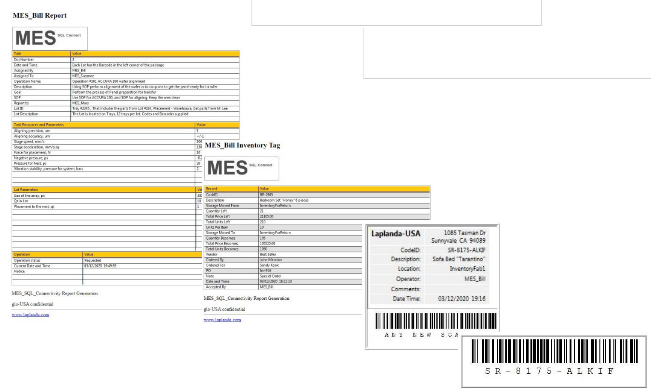
MES_SQL_Connectivity Data Types	Database Connectivity Toolkit Data Type
Double numeric	double
Boolean	string
String	string
Date/Time string	date/time
Time stamp	date/time
Path	string
I/O channel	string
Refnum	binary
Complex numeric	binary
Extended numeric	binary
Picture control	binary
Array	binary
Cluster	binary
Variant	binary
Waveform	binary
Digital waveform	binary
Digital data	binary
WDT	binary
Fixed-point numeric	binary

GUI


MES SQL Connect
MES SQL Connect
Login as
Login Password
MES
EXIT


INVOICE
Lapalnda-USA
SALES RECEIPT
PURCHASE ORDER
PACKING SLIP


MES SQL Connect
Login
Work
Task
Operations
MES SQL Connect
Inventory List
Inventory Move To
Refresh
Not enough Quantity
OK
EXIT


MES_Bill Report
MES_Bill Inventory Tag
Lapalnda-USA
SR-8175-ALKIF