**Introduction:** Objectivity/DB is a key component of modern object oriented product solutions for the Next Generation network. It enables rapid development of Element Management Systems [EMS] and is a very valuable tool for development of real-time Network Elements [NE]. Its ability to store and process complex structures (collections, trees and networks) make it the ideal choice for new applications across the whole spectrum of communication domains, for example:

- Wireless PCS - The Ericsson (Qualcomm) CDMA Intelligent Base Station Controller and the Nortel CDMA Base Station Controller (used in SprintPCS).
- Low Earth Orbit satellite systems (Iridium).
- Broadband – Marconi’s high throughput ADSL equipment.
- DSL - Catena Networks enables mass-market deployment of broadband DSL services.
- Wireless Broadband – Navini Networks manufactures non-line of sight high-speed data equipment.

This White Paper examines the technical challenges faced by designers of an advanced EMS. It contains technical case studies describing how they used Objectivity/DB to overcome these challenges and achieve faster, safer deployment of new applications. It then briefly outlines the pertinent features of Objectivity/DB and outlines material that you can use to evaluate the suitability of Objectivity/DB for your project.

**The Technical Challenges:** This section considers the technical challenges and opportunities facing an EMS designer. It considers existing and Next Generation networks and suggests increasing product value by making distributed EMS and Network Element (NE) applications work together, using a common object model.

**Telco, IP and NextGen Networks:** As voice and data networks merge into the network of the future, management of network elements will play a stronger role in network operations. Rapid introduction of new features requires that EMSs be an integral part of the network product offering. Telecom operators and equipment manufacturers will move rapidly to introduce new media services. At the same time, data network operators are striving to migrate data networks to be more reliable and be capable of
providing real-time voice and video services.

Next Generation network solutions must be agile and need to respond rapidly to new technologies and network architectures. The Network Element and the EMS together form a cohesive and valuable product offered by equipment providers. The two systems, exploiting distributed computing, can provide a very cost-effective solution to customers and reduce development time and cost for the equipment supplier.

Making the Element Management System and the Network Element Work Together to Increase Product Value: Deploying Objectivity/DB on the Network Element and the EMS offers the equipment supplier a unique opportunity to increase the overall value of the product offering while increasing quality and reducing the development time and costs. With Objectivity/DB, Network Elements and the EMS work together to solve the complex communications problems of the next generation product line. The Network Element and EMS may share the same database schema, object methods and implementations. This obviously reduces effort and development time and harmonizes the two systems.

The Element Management System’s relationship to the Network Element: The EMS has a very valuable role in the overall product offering. It is the primary point of contact for the customer and how the Network Element is viewed and managed. Flow through provisioning and alarm reporting are two major functions the EMS provides on behalf of the Network Element.

The TMN architecture specifies five functional areas of responsibilities to the EMS, listed in the table below. Objectivity/DB can play an important role in implementing many of these functions in a very cost-effective manner. All of the functions are tightly associated with the actions and responsibilities of the Network Element. It is obvious that the EMS has a great deal in common with the Network Element.
Deploying Objectivity/DB In Element Management Systems

**Fault Management**
- Alarm handling
- Trouble detection
- Trouble correction
- Test and acceptance
- Network recovery
- Alarm forwarding
- Filtering
- Log Management
- Diagnostics

**Capacity Management**
- System turn-up
- Provisioning
- Autodiscovery
- Backup and restore
- Database handling
- Inventory management

**Accounting Management**
- Track service usage
- Bill record management
- Service level agreements

**Performance Management**
- Data collection
- Report generation
- Data analysis
- Performance Monitoring

**Security Management**
- Control NE access
- Enable NE functions
- Access logs

**Common Object Model**: By sharing object models and design, reuse can be maximized. This reduces cost in development, testing and maintenance. A common product object model facilitates long-term product support. Change management is more effectively managed when a uniform view of the product is readily available.
A common object model for the system, Network Element and EMS would contain classes such as Alarm and Fault for each component part of the Network Element. For example a Network Element may have two power supplies which are normally both active, although one is sufficient to provide all required power should the other fail. The Network Element may detect “low voltage” on one supply, raise an alarm, and inform the EMS. The design and implementation could be shared between the Network Element and EMS. The EMS could extend the class for persistence and store the alarm in an Objectivity/DB alarm log database.

Many network operators are required to report certain service outages to the FCC. The FCC analysis of these outages over time reveals that most outages occur during software upgrades of Network Elements. This severe problem can be addressed using Objectivity/DB. With Objectivity/DB the object model is the database schema. When the EMS shares the Network Element database schema the EMS can provide added value to the product offering. When new features are added both the Network Element and the EMS object definitions are changed and new releases are easier to co-ordinate.

With the Objectivity/DB schema evolution feature, the EMS can easily migrate its database to the new schema. When the Network Element database is a part of the EMS each Network Element database can also be migrated forward at the EMS. This migration can be done prior to upgrading the Network Element with the new software release. Should any potential service affecting error occur during the migration it could be reported by the EMS and corrected prior to deploying the new release to the Network Elements. When the Network Element software is upgraded, the new database can be downloaded to the Network Element. The Network Element does not have to be burdened with the task of moving its database forward to the new schema. This greatly reduces the amount of processing done at the Network Element during upgrades, reduces upgrade time and reduces the possibility of a service outage. Reduction of service outages means improved customer satisfaction, reduced human intervention and reduced reporting to the FCC.

There are many other advantages to using shared schemas. One of the
major functions of the EMS is to provide a database backup for all Network Elements. Since the EMS has a backup of the active Network Element database, and can make queries and reads to the backup database, the number of messages sent to the Network Element can be reduced.

By sharing schema between the EMS and Network Element, many other functions currently done by the Network Element can be assumed by the EMS. This reduces the development and testing burden on the Network Element. It can increase reliability of the product line in general and greatly reduce the time to market.

Objectivity/DB in the Element Management System: Objectivity/DB is a feature rich component of the EMS and offers many advantages to the equipment supplier. Object oriented design methods are used in the design of all modern EMSs. Objectivity/DB provides design and implementation consistency of persistent objects. Objects designed in UML can be made persistent without shifting paradigms and tool sets. What you design is what you get. This design consistency can reduce development time, errors and defects. Using Objectivity/DB can reduce database code within the EMS by as much as 40-60%. It allows the engineer to focus on revenue generating features and less on mechanics. This White Paper will prove that Objectivity/DB also provides very significant run-time performance advantages over a legacy storage scheme.

Objectivity/DB is highly scalable in the dimensions of data volume and processing elements. Data intense functions such as performance, billing, and alarm management may require very large databases. Objectivity/DB has demonstrated this with large-scale database deployments. Objectivity/DB also can scale to N-tier processing architectures with replicated or segmented database within a computing cluster. It provides transparent object persistence within J2EE architectures.

The EMS must exist in a protocol rich environment. That is, many protocols must be supported between the EMS and the Network Management Layer; they can be SNMP, TL1 or CORBA. Likewise, there may be standard or proprietary protocols to the Network Element. All the protocols support the fundamental containment construct within network management. Each
Network Element is made up of component parts, which in turn are made of smaller parts - all are managed objects.

This notion of containment is ingrained in network management. With most of the protocols, there is an associated Managed Object repository. The message format and the Managed Object repository are designed based on the containment paradigm, which is also a foundation of object orientation. It is counterproductive to try to represent these natural objects in any other form. The components can be persistently stored within the Objectivity/DB in the most convenient object form. Objectivity/DB enables traversing persistent trees, networks and similar objects. This basic component aggregation paradigm is very difficult to map to non-object databases.

**Objectivity/DB in the Network Element:** When the EMS and Network Elements share a common object model with persistence services provided by Objectivity/DB, many issues can be better addressed within the EMS. Objectivity/DB also provides great benefit to the Network Element itself. Most Network Elements have stringent availability requirements and must operate with five 9s reliability. For this reason, all Network Elements employ some redundant processing architecture. Main processing cards and other critical components have a dedicated backup unit or use an N+1 scheme of fail over. In these architectures, provisioning and other service critical state information must be replicated in the backup units. Replication is a very difficult problem to solve and test. A very large portion of the engineering effort of the Network Element is focused on replication and fail over recovery. When Objectivity/DB is used at the Network Element, this effort can be greatly reduced. Objectivity/DB offers replication persistence within hard real-time operating systems environments. Replication of persistent objects can be achieved without additional development by the engineering staff. Replication of the objects is important but of equal importance is recovery of a failed unit. When a processing card recovers from a failure the database within must be verified and be made current with the state of the current processing elements. This can be a difficult problem and can result in recovery failure should the state information be inconsistent with the active element. The Objectivity/DB replication services can greatly reduce the cost of development and testing, improve quality, and reduce outages.
Deploying Objectivity/DB In Element Management Systems

**Distributed Environment**: By taking a system wide view a holistic object solution can be developed to support current and future product offerings. The ideal object design spans both the EMS and the Network Element with real-time, critical service delivery functions deployed on the Network Element with common reporting and provisioning functions deployed on the EMS. This approach leads to a better design that reduces redundant work and maximizes reuse. Using Objectivity/DB as the database on both the EMS and the Network Element enables large gains in software engineering productivity.

**EMS Case Studies**

**Case Study 1 - Wireless CDMA**

**Operation, Administration and Maintenance**: The OAM&P BSS Manager provides the Operations, Administration, and Maintenance (OA&M) interface for the CDMA system RF infrastructure – the Base Station Controller (BSC) and Base station Transceiver Subsystem (BTS).
The BSS Manager is a UNIX-based system administrative terminal for the CDMA RF network. It allows the operator to configure, monitor, and troubleshoot various CDMA network elements. It logs, reports, and manages alarm events from its managed subsystems.

The Common Management Information Protocol (CMIP) defines agent-manager messaging. The schema is defined according to the specification of the Generalized Definition of Managed Objects (GDMO).

BTS: Base Transceiver Subsystem
BSC: Base Station Controller
BSS Mgr: OAM&P BSS Manager - A single logical view of the system.
SS: Subscriber Server - Can be on one machine or can be distributed on multiple machines. A redundant SS Server can manage the same set of BSCs in failover mode.

Case Study 2 – Cable Modems
Network Management and Provisioning System (NMAPS)
System overview: The application allows Management and Provisioning of the system through the use of three data groups, the Topology database, which represent the physical state of the network, the Alarm database, which stores the recent alarm information, and the Polled Statistical database, which contains information polled from the hardware periodically.

The object model represent the network of cable modems (STUs) and their hub controllers (HCXs). It also includes the individual subscribers and their information. The relationships between subscribers, modems, and statistical data are represented in the database schema.

The GUI network management applications display information and support various types of queries. The management application has a single logical view of the distributed databases and hence of the whole system.

The system must support 100,000 STUs, generating large amounts of statistical data per day. Several daemons populate polled data into their databases at 15-minute intervals. Other daemons service requests and events against the Topology, Alarm and Polled Statistical databases.

Case Study 3 – DSL
Deploying Objectivity/DB In Element Management Systems

**DSL EMS:** This Objectivity customer currently offers two Systems that include an EMS built on Objectivity/DB – a Broadband ADSL system and a Broadband Loop Carrier system. The ADSL product enables service providers to deliver POTS and ADSL services on any copper pair, without reducing the number of available POTS lines. The Broadband Loop Carrier (BLC) is a new class of access system, which will enable carriers to accelerate DSL service delivery, simplify the access network and smoothly migrate from today's circuit-based PSTN to a converged, packet-based public network. By integrating POTS and DSL on every line, at POTS economics, the new product helps service providers dramatically reduce their capital and operations costs. With BLC, every subscriber line will support lifeline telephone service and will be "DSL ready" the moment it is installed. All operations, provisioning and maintenance can be performed remotely, without truck rolls to remote sites.

To support the provisioning and management of mass-market DSL services, the new equipment utilizes a robust DSL EMS and complete Application Programming Interface (API) suite. The EMS is expandable to tens of thousands of lines and can be distributed across client/server platforms for optimum performance and reliability. The full-featured API suite provides the interfaces necessary to electronically link the new equipment to upstream Operation Support Systems for flow-through provisioning and reporting. The EMS has a CORBA API that facilitates seamless integration into northbound NMS and OSS legacy systems using industry standard technology. Service providers are able to use the CORBA API to achieve true ADSL flow-through provisioning right from the Network Operations Center down to the Network Element. The EMS Central Server, MOA Agents, and the Client GUI can reside on the same hardware platform.

- The system has a single logical view of the data (one federation).
- The MOA agents work with the MO and keep the database in sync with the hardware status. The MOA agents implement the Fault, Configuration, and Performance modules from the FCAPS. They can control multiple MOs.
- The Central Server implements the Administration and Security modules.
CASE STUDY 5 – Optical Networking EMS

Element Manager and Network Control Manager

The EMS product is an element manager that provides functions at the Element Management Layer and support to the Network Management Layer. It performs configuration management, fault management, performance management and security management. The EMS manages ranges of optical systems plus SDH multiplexers and cross-connects. It can be used stand-alone or in conjunction with a Network Control Manager.

The Network Control Manager provides functions at the Network Management Layer and support to Service and Business Management Layers for the full range of Dense Wavelength Division Multiplexing (DWDM) and Synchronous Digital Hierarchy (SDH) equipment.

A combined product integrates the Network Control Manager and EM-OS element managers, allowing the management of the whole customers' optical network from a single point of access. It is able to create end-to-end circuits, executing all necessary physical configurations without user intervention. The product performs the network layer functions of trail manage-
ment, network-wide alarm indication, trail auto routing, dynamic trail protection and performance management.

The graphical user interface provides a user-friendly environment for the operator. The system is hosted on a scaleable UNIX-based server and can manage networks with 1000s of network elements and many element managers. The product supports any combination of ring, point-to-point, chain or star network topologies. Network elements can be grouped into sub-networks. The product can also create and manage Virtual Networks.

Objectivity Advantages

<table>
<thead>
<tr>
<th>Feature</th>
<th>Wireless CDMA EMS</th>
<th>Cable Modem EMS</th>
<th>DSL EMS</th>
<th>Broadband Wireless EMS</th>
<th>Optical Networking EMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex Object Model</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Data Clustering</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Distributed Architecture</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>High Availability</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Low administration</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Lower cost of ownership</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MROWs</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Performance</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Availability</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Schema Migration</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Small Footprint</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Time to market</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Complex Object Model – The complex EMS model is easier to map into Objectivity/DB and this allows very fast navigation to system data.
Data Clustering – Objects are grouped in databases held close to the controlled NEs.
Data Replication – An element’s data may be replicated on another node, allowing the system to function in the case of an agent failure.
Distributed Architecture - Allows data to be located closer to the point of use, with a single logical view of the system for all system components.
High Availability - The system employs fault-tolerant hardware and Objectivity/DB to help ensure non-stop operation.
Low administration – “Close to Zero” administration is a main requirement.
Lower cost of ownership - Ongoing SW maintenance costs tend to be very high especially if the EMS supports a complex object model.
MROW - Multiple Reader, One Writer mode provides the highest possible level of concurrency.
Performance - The performance of the EMS is essential to enabling it to manage an economical number of NEs. Using Objectivity/DB allows the system to accomplish this performance goal with very lightweight require-
Deploying Objectivity/DB In Element Management Systems

- Scalability: The system will grow in size, and the application will be able to handle this growth by adding systems and databases to the federation with no changes to the system configuration.
- Schema Migration: Easy software and data migration is crucial in a fast-growing market.
- Small Footprint: Objectivity/DB is completely transparent to the user since it is embedded inside the product.
- Time to Market: EMS software availability and flexibility are critical factors, especially the ability to develop new features rapidly.

Objectivity/DB Meets The Technical Challenges

High Performance with Trees and Networks of Objects: High performance is critical in network element management applications. The object models are often highly complex tree or network structures. Object models expressed in ASN.1 and UML can be imported into Objectivity/DB via tools such as Rational ROSE and ROSElink for Objectivity. XML models (and
Data) can be imported and exported with Objectivity tools. Objectivity/DB can directly represent and manipulate standard C++ and Java collection classes, such as trees, sets and lists. The Scalable Collection Classes are a standard part of the C++ and Java APIs. Objectivity/DB also supports persistent versions of the C++ Standard Template Library collections.

This diagram shows a simplified model representing the MIB tree structure (top left) and the Managed Object (MO) relationships. The MIB classes contain the description of the data and the MO class instances contain the correct/concrete values for a specific system. Inheritance increases the flexibility of the EMS. Direct representation of the relationships increases navigational performance.

Navigation & Inheritance Mechanisms

Faster Navigation - In this example, an RDBMS query that needs to find all of the Alarms associated with an Element would perform 2 * N accesses to the B-Tree indices (where N is the number of Alarms) and 2 * N logical [row] reads into the Join table and the Alarm table. Objectivity/DB only needs 1 B-Tree access and (1 + N) logical reads.
Deploying Objectivity/DB In Element Management Systems

**Inheritance** - As the inheritance class hierarchy grows deeper the relational database has to create more tables and execute more JOINs, so performance degrades. Objectivity/DB handles inheritance deep within its kernel to ensure high and consistent performance.

**High Performance Clustering and In-Memory Caching**

Clustering reduces the number of physical I/Os needed to access a group of related objects. The application designer can cluster objects that are generally accessed together into a single logical “page”. Reducing the number of I/Os needed to service a transaction can dramatically shorten transaction times and increase overall system throughput.

Caching reduces I/Os by attempting to keep frequently used data in RAM; or by reusing RAM immediately if data is being streamed to or from disk. The critical portions of a MIB may be locked in a large RAM cache across transactions. Alarms may be streamed to disk through a small RAM cache.

In Objectivity/DB the clustering and caching strategies are defined at runtime on a per-thread basis, unlike RDBMSs that rely on a database administrator to control server behavior.
High Performance with Concurrently Accessed Objects

MROW [Multi-Reader One Writer] - Many algorithms examine large amounts of reference data while processing new data. MROW provides readers with a consistent view of committed data while one writer is updating that set of data. MROW is more effective than fine-grained locking because it eliminates the need to repeatedly set and release read locks.

IPLS [In-Process Lock Server] - Most network element management systems deploy multitreaded servers in dedicated processors to handle the most demanding applications. The IPLS can be directly linked with the application “server” process in order to avoid TCP/IP and network overheads. Applications that are able to use the IPLS have shown throughput increases of 20 to 40 percent.

Distributed Architecture

A Federated Database presents a single logical view of all of the databases and the accompanying schemas [object definitions]. The distributed architecture is fundamental to Objectivity’s fault tolerance and data replication capabilities. It offers new degrees of freedom in architecting systems to acquire, store, process or access information at any layer of the telecom infrastructure. In a system that embeds Objectivity/DB there is no longer a compelling need to push data back to a central server; or to build supplementary file caches close to or within the real-time equipment. Data can be stored and processed close to the point where it is most often used, but it (or
a replica) can be made accessible to any authorized user or subsystem anywhere in the network.

Data Scalability

**Single Logical View** - Each object is uniquely represented by a 64-bit composite Object Identifier [OID]. Each component of the OID is a logical number, not a physical identifier, allowing Objectivity/DB to dynamically relocate objects to reclaim physical space quickly. Federated Databases can scale into the Petabyte [1000 Terabyte] range.

Groups of objects may be clustered within containers (loosely analogous to a file) that are clustered within databases. The databases are grouped within a federated database. Objects may be directly associated with other objects by using named 1:1, 1:many, many:1 or many:many links. The links may span containers and databases, allowing the creation of huge networks of objects spread across databases held on many machines. Navigation across databases is completely transparent to the application.
Mass Storage Integration - Many products have loose or single vendor integration with hierarchical mass storage devices. The Objectivity/DB remote data server automatically integrates with most standard file server products. In 1998 Objectivity pioneered the integration of a database with the High Performance Storage System (HPSS), a technology that resulted from a joint project between industry and the National Laboratories. The Objectivity Open File System (OOFS) layer integrates the Objectivity/DB remote data server with HPSS. The OOFS classes may be modified to interface with other non-standard storage devices. Customizable security (e.g. using Kerberos) can be invoked via a Generalized Security Architecture interface implemented behind the Page Server interface. The largest deployment of this technology currently holds over 500 Terabytes of objects, growing at over a Terabyte per day.
Continuous Availability

Objectivity provides continuous availability through a combination of features that work together to minimize downtime due to system maintenance, application upgrades or faults in hardware, network or system software.

On-line administration - Objectivity/DB allows full and incremental backups to be performed without interrupting applications. Other tools permit movement of databases and on-line partitioning and repartitioning of federated databases across platforms.

Active Schema Option allows applications to access, change and add object class definitions. It even allows C++ applications to create new classes and start manipulating object instances without compiling code. This can add considerable flexibility to the configurability of systems to suit local conditions.

Fault Tolerant Option is designed to guarantee database availability in networked environments, where databases are distributed across computers. Faults at a database site are isolated and do not affect other databases. The Fault Tolerant Option is complementary to hardware solutions implemented with fault-tolerant processors and redundant disk (e.g. RAID).

Data Replication Option (DRO) permits on-line creation and deletion of database replicas. Made possible by Objectivity's underlying distributed architecture, DRO represents a dramatic advance in replication technology based on academic research performed in the early 1990's. Dynamic quorum (majority vote) calculations guarantee system integrity without constraining write access to one database master. To guarantee data integrity across replicas, the database manager performs a quorum calculation when it accesses a database. Portions of the federated database disconnected from the quorum are resynchronized with the quorum upon reconnection. As an example, data replication used inside a routing application can provide hot-failover in the case of platform failure and automatic resynchronization of data after the problem is corrected. Data replication across a WAN allows one time delivery of processed data to back-office applications or a carrier's partners.
Objectivity/DB – Key Features

The architecture of Objectivity/DB was designed to provide the highest levels of reliability, performance, scalability (data volume and concurrency), distribution and interoperability. It provides the following major benefits to EMS designers and developers:

• An open but safe long-term repository for valuable data.
• Considerably simplifies the design, development, testing and deployment of complex network element management systems because:
  • It has the most powerful object modeling, storage and access capabilities of any Object Database Management System.
  • It provides a complete storage solution for metadata, Management Information Base data and generated data, such as alarms.
  • It can navigate trees (such as those found in a Management Information Base) and networks of objects much faster than relational or hybrid object-relational database management systems.
  • Removes the bottlenecks inherent in object servers. It reduces
Deploying Objectivity/DB In Element Management Systems

network traffic and employs smart caching to avoid redundant disk I/O. It can be deployed anywhere in the network, even in the real-time elements.

- Outperforms persistent language, message based and home grown file-based systems as data volumes and concurrent usage increase.
- Objectivity/DB is better at making the location of data, metadata and DBMS processes transparent to client and administrative processes than any other DBMS.
- Distributed architecture for networked and multithreaded or multiprocessor environments.
- Objectivity/DB Open File System (OOFS) provides seamless integration to a wide range of industry standard hierarchical storage managers, Storage Area Networks, Network Accessed Storage and HPSS [High Performance Storage System] nodes. This opens up new product opportunities for archiving and data mining service information, e.g. for long term correlation and identification of the probable cause of transient alarms.
- Complete interoperability between all supported platforms [UNIX, Linux, Windows NT/2K and LynxOS] and languages [ODMG’93 C++, Smalltalk & Java and ANSI SQL & ODBC].
- Tools for importing and exporting XML class definitions and data. Objectivity Active Schema™ and Objectivity ROSElink™ can be used to build schema management tools or to migrate UML or GDMO definitions in and out of Objectivity/DB.
- The Fault Tolerant Option replicates system data and services. The Data Replication Option replicates user databases to multiple sites. Applications can survive both network and node failures in geographically or logically distributed systems.
- Objectivity/DB is built to cope with the realities of environments that include rapidly evolving applications and technologies. The Active Schema™ option supports the dynamic definition, creation and migration of persistent C++ or Java object instances.

Objectivity/DB Meets The Business Challenges

Faster, Safer Deployment: The performance, reliability, flexibility, interoper-
Deploying Objectivity/DB In Element Management Systems

rability and scalability of Objectivity/DB undoubtedly make it the best data-
base for new Element Management Systems. Its powerful object modeling
and manipulation capabilities eliminate the need for the mapping layer
needed by legacy DBMSs. This reduces software development and quality
assurance effort and time to market.

Technical excellence is only a part of the solution in today’s dynamic techni-
cal and business environments. Other factors to include in the choice of a
database partner include:

**Adherence to Standards** - Objectivity is an active participant in standards
activities related to telecom systems, notably the Telemanagement Forum.
Objectivity/DB uses standard languages, compilers, operating systems and
database APIs to reduce the learning curve for designers and programmers.
Domain expertise - Objectivity has a large user base of software and
hardware vendors that embed Objectivity/DB within their network element
management products.

**Availability of Expert Help** - Objectivity can provide expert consulting at the
critical points in the design and deployment of a system. Objectivity’s
expertise has been gained by helping hundreds of customers, ranging from
small, embedded systems to the deployment and maintenance of extremely
large production databases. We can help reduce your project’s risks and
your product’s “Time to Market”.

**Lower Cost of Ownership**: Eliminating mapping reduces costs by as much
as 30% - Objectivity/DB is not only faster at navigating trees and networks of
objects than legacy technologies, it also reduces the amount of engineering
effort needed to bring a software product to market. The cost savings that
result from eliminating the mapping layer between the programming
language and the DBMS can be as much as 30% in a typical database
project.

**Zero Administration Philosophy** - Traditional DBMSs were designed to be
managed by highly trained database administrators. Objectivity/DB was
designed with a “Zero Administration” philosophy, making it much easier to
embed it within a product. Its libraries, object definitions and pre-loaded
databases can be shipped with your product and installed by customer staff with no DBMS expertise.

**More Efficient Hardware Utilization** - Objectivity/DB has a much lower memory and disk footprint than legacy DBMSs. Each client typically requires between 1.2 and 3 Mb of RAM for the shared library code, plus whatever cache the application needs for the objects it is working on. The lower footprint and the more efficient use of processor power translate into greater throughput on an existing hardware configuration; or less expensive hardware for new installations.

**Increasing the Value of the Data Asset** - Any revised or new application running on Objectivity/DB will automatically add powerful new features to your product with very little extra effort on the part of your existing engineers. It will also make it easier to handle system upgrades and language or platform changes in the future. Your user’s data will be protected, be more accessible (with appropriate security) and be made more valuable as it accumulates over time.
Summary

We have seen that Element Management Systems make many demands on the underlying database management system:

- High Performance with complex data – algorithms need the data at RAM speeds, not disk speeds. Data correlation systems may need to deal with batches of data or continuous streams from multiple sources.
- Scalability - in both data volume and the number of concurrent users (or threads).
- Reliability – network element management systems are deployed in “five nines” and “six nines” environments.
- Data Distribution – data may have to be globally available. It may be cached at one or more geographic sites to reduce the load on the carrier’s management network
- Interoperability – data may be captured and processed on different kinds of equipment. It may be processed in C++ and accessed via tools built with Java; or generated on an RTOS, processed on a UNIX/Linux box and viewed on a Windows PC.

Objectivity/DB meets these demands by providing:

- Higher, scalable performance in a wide variety of centralized and distributed configurations.
- High reliability in 2-processor or multi-processor, multi-platform deployments.
- Additional functionality at a very low cost.
- Faster time to market because of the elimination of mapping code and its powerful object modeling and manipulation features.
- Lower cost of ownership because it is easier to maintain, upgrade and expand deployed applications and systems.

Objectivity/DB meets these demands without imposing extra burdens on the designer, programmer or user. Objectivity/DB is an invisible, silent partner that strengthens a product and helps bring it to market with less effort.
Deploying Objectivity/DB In Element Management Systems

Additional Resources

Other White Papers
- Objectivity/DB in Telecommunications Applications
  http://www.objectivity.com/DevCentral/Products/TechDocs/pdfs/WPTELCO.pdf
- Choosing a Higher Performance Database
- Using an Object Database in Intelligent Network Applications
  http://www.objectivity.com/DevCentral/Products/TechDocs/pdfs/IntelligentNetworkWP.pdf

Technical Overview of Objectivity/DB
http://www.objectivity.com/DevCentral/Products/TechDocs/TechOv.html

Trial Product Download