Eric A. Newcomer

Multivendor Integration Architecture: Standards, Compliance Testing, and Applications

The Multivendor Integration Architecture (MIA) is a user-driven initiative that addresses the practical application of open systems software standards to business requirements. This paper provides historical background and context for this standardization effort and describes Digital's contributions to the effort, particularly in the area of distributed transaction processing. Digital complied with the MIA specifications, integrated compliant products into a complete platform, and delivered a large application on the platform.

In today's competitive environment, an enterprise's computer systems help determine its success or failure. The need for large enterprises to separately manage applications on different computer vendors' platforms distracts the enterprises from performing their main business functions and adds to their operations cost. Corporate mergers and acquisitions often compound the problem.

While the business need for high-quality computer systems has never been greater, established computer users find themselves in a poor position due to the tremendous burden of their legacy systems. Newer companies almost automatically gain a competitive advantage from their more flexible, state-of-the-art computer systems.

The availability of open, standards-based systems enables critical business systems to be built on a common platform that can be purchased from multiple vendors at competitive prices. This offers everyone the same level of basic functionality with which to build new systems. These systems must be capable of integrating components from multiple vendors into a single, large application.

This paper provides background information for user-driven standardization efforts, with a focus on Nippon Telegraph and Telephone's (NTT's) Multivendor Integration Architecture (MIA). The paper discusses the MIA's principles, including three multivendor interfaces, NTT's major types of computer processing, specification development, and Digital's approach to addressing integration problems related to transaction processing (TP). Also discussed are implementation and systems integration issues and the delivery process. Digital's contributions to the open systems software integration effort are described. Digital was instrumental in defining the MIA specifications for TP, and it developed the first MIA-compliant application.

User-driven Standardization Efforts

About 25 years ago, NTT, one of the world's largest corporations, developed its first computing system procurement specifications. These detailed specifications included designs for special hardware and operating systems to meet the enterprise's demanding requirements.

The procurement specifications focused on systems of sufficient capacity and robustness with which to automate the fundamental business operations of a large telephone company. They did not require portability or interoperability. NTT presented the specifications to Hitachi, Fujitsu, and NEC and ordered hardware and software that conformed. In addition to the Japanese suppliers, IBM also responded to the procurement request and became an NTT supplier.

Following the successful implementation of the original specifications, NTT developed applications on top of the various vendors' platforms. Like many other large enterprises, NTT created separate teams to tackle the vendors' systems individually.

In 1988, NTT established the MIA consortium to resolve the inefficient practice of having separate teams develop and manage applications on different vendors' platforms. The consortium was charged with addressing the associated problems that interfere with the way these applications communicate, share code, share data, or move to a new technology base.

The MIA initiative was conducted as a Japanese industrial collaborative research project with the goal of resolving the problems of multivendor application environments. NTT invited computer vendors to join the project by issuing a public subscription announcement and then selected participants from among the respondents. Fujitsu, Hitachi, NEC, and IBM were the first consortium members. Digital was also selected because of its expertise in networking and client-server computing. The MIA initiative set out not only to resolve the problems with a multivendor environment but also to move NTT's computing systems forward by incorporating distributed processing functionality.

One of NTT's goals was to eliminate all visible differences among the vendors' platforms. "Visible" meant perceptible to (1) the humans who interact with the computers as end users, in application development and deployment, in system administration, and in network configuration and management, and (2) the protocols for communication between the different vendors' computers. A guiding principle of the MIA initiative was that the systems with which people interact should appear identical, regardless of the manufacturer who created the hardware or software being used or the purpose for which it was being used.

As a member of the MIA consortium, Digital helped develop detailed specifications that met NTT's requirements for open systems software components that any vendor could implement. In particular, Digital developed new multivendor specifications for distributed TP, an area of computing for which standards did not exist.

The results of the MIA project were published in 1991 as 11 volumes of detailed procurement specifications that describe a complete application development platform for large-scale systems.[1] Applications created using software that conforms to the specifications can be developed and implemented on any vendor's computer.

The concepts behind the MIA specifications were put to the test at a public demonstration at Interop Tokyo in July 1994. After considerable debugging and testing, the concepts were proven to work.[2] The next measure of success is whether sufficient demand and cost savings exist to induce vendors to market conforming products, in particular, off-the-shelf products.

Digital's involvement in specifying solutions to user-driven open systems software requirements continues at the Service Providers' Integrated Requirements for Information Technology (SPIRIT) consortium, which is sponsored by the Network Management Forum. SPIRIT members include the world's largest telecommunications service providers and computer vendors. The MIA specifications were submitted as base input documents for SPIRIT, along with other documents from AT&T, Bellcore, BT, and ETIS (a consortium that represents 27 European postal, telegraph, and telephone administrations).[3]

It is unknown whether this user-driven approach to standardization will succeed and meet the important goals of portability, interoperability, and multivendor procurement.[4] Nonetheless, users and vendors are learning some important lessons as a result of the users' strong efforts in this area.

MIA Principles

When NTT turned its attention toward creating the MIA procurement standards, it began to attack the problem of multivendorization, which NTT believes is strategic to its future business. "Because a computer system must be able to provide as broad a range of business services as possible, it is desirable to construct such a computer system flexibly enough to include different computers, each of which covers the area of business in which the vendor's model is the most powerful."[5]

Early in the MIA project, NTT established the basic requirement that solutions be based on open systems standards where possible. However, since the corporation's existing complex legacy of applications was critical to business operations, the new standards had to allow for the same degree of functionality and robustness as the software for the existing platforms. Also, if it was to replace its current applications with applications that took advantage of commodity technology, NTT needed a way to migrate to the new while interoperating with the old. "Based on the assumption that a variety of hardware and operating systems of vendor-specific design is widely accepted in the general-purpose computer market, MIA specifications must be a feasible extension of, and coexist with, vendor-specific architectures."[5]

The MIA effectively grouped related functionality to match the existing requirements for business applications and added support for distributed client-server computing. Using the resulting architectural framework, the MIA consortium matched existing standards to NTT's needs, identified missing functionality, and created new multivendor specifications to achieve the additional functionality.

Three Interfaces

At the start of the MIA project, NTT identified what it considered the three most important issues of multivendorization:

1. Duplicated development of application programs
2. Difficulties in resource sharing
3. Differences in operating methods[5]

For each of these problems, NTT identified solutions in terms of standard, i.e., multivendor, interfaces, as follows:

- Application portability using standard application programming interfaces
- Interoperability using standard communication protocols
- Common user interface using a windowing style guide

Figure 1 illustrates the basic architecture as specified by the MIA consortium. The configuration incorporates three systems—the end user, the departmental computer, and the host computer—and includes three types of interfaces—human user interface (HUI), application programming interface (API), and systems interconnection interface (SII). The figure represents the fundamental goal of MIA conformance for each vendor, i.e., to offer conforming interfaces and protocols that allow NTT to purchase the same level of compatible software functionality from multiple vendors and create new applications that are inherently distributable, portable, and interoperable. Another reason NTT focused on these three interfaces was that if the MIA specifications contained too many low-level interfaces, the vendor-specific strengths would be removed and the specifications would not support the NTT strategy of multivendorization.

Through the standardization of the three interfaces, NTT anticipated that an end user would be able to use any display device without knowing the vendor (via the HUI), a programmer would be able to write a program that would run equally well on all platforms (via the API), and a computer from one vendor could be connected to a computer from any other vendor using common systems interconnection protocols (via the SII).

Additional types of interfaces and protocols that were outside the scope of the MIA specifications are being addressed by the SPIRIT consortium. For example, SPIRIT has taken on the task of standardizing the system management interfaces and protocols. At the start of the MIA initiative, NTT decided that the best use of time and resources would be to standardize the HUI, the API, and the SII.

Major Types of Computer Processing

NTT categorized its computing activity into four types: real-time processing, transaction processing, interactive processing, and batch processing. Figure 2 illustrates the processing types and interfaces addressed by the MIA specifications. Note that the specifications did not address real-time processing issues.

NTT included the area of TP because the company had a huge investment in developing and running TP systems and because its business relied on TP systems such as billing, inventory control, and directory assistance. The opportunity for return on investment was therefore high for this critical application area. Data

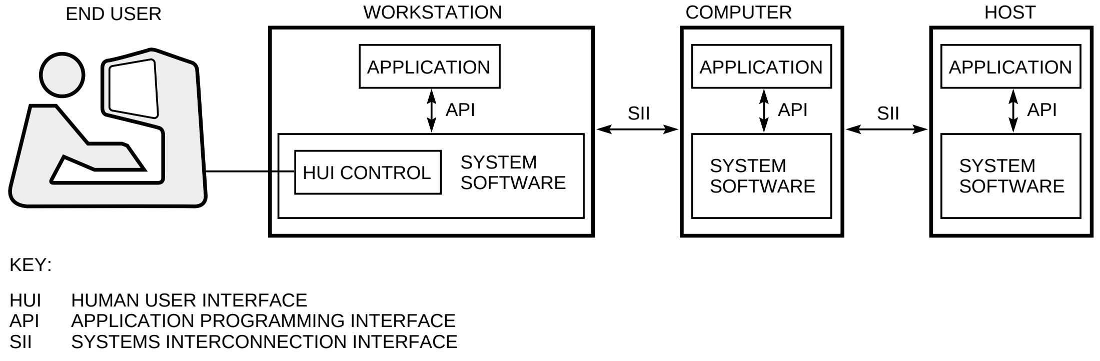


Figure 1
MIA System Configuration

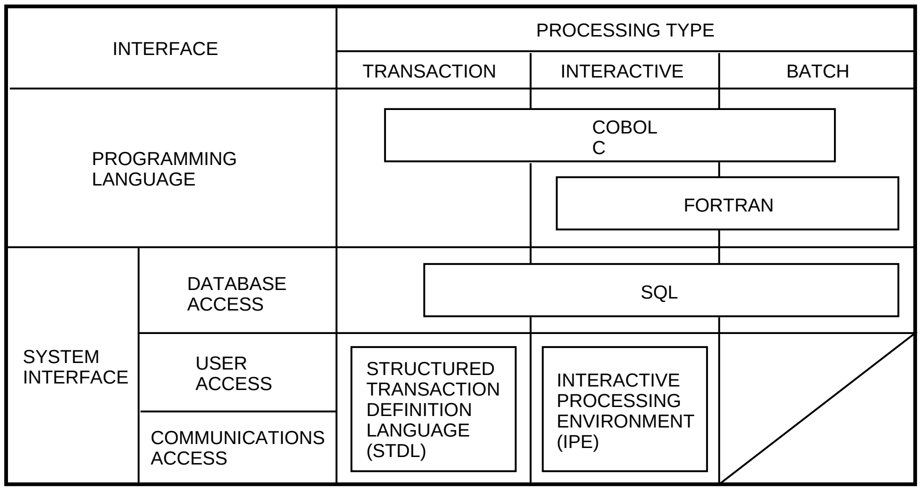


Figure 2
MIA Processing Types and Interfaces

integrity, remote access, and system reliability are the key characteristics of TP that needed to be supported through standards compliance to fully realize the cost savings potential of the MIA.

In the area of TP, no international standards existed for the two most significant interface areas NTT had identified as candidates for multivendorization: the API and the SII. This deficiency created one of the biggest problems that the MIA consortium had to resolve and later gave rise to a large systems integration and application delivery challenge with respect to the MIA.

NTT required the MIA TP specifications to support remote, distributed transactions. MIA TP comprised specifications for multiple programming languages and network protocols and therefore became the widest integration point that had to be achieved.

Developing the Specifications

As the first step in specifying solutions to the problems that it put forth to the MIA consortium, NTT produced user requirements. The user requirements evolved over the course of the project as new questions arose that needed clarification from NTT's business sector. Meeting user requirements was the final verification of the specification output at the end of the project. In addition, the consortium had to develop specifications that could be implemented by any vendor.

For the area of TP, NTT asked each vendor in the MIA consortium to submit a proposal for a new multivendor specification and selected Digital's Application Control and Management System (ACMS) TP monitor proposal as the base on which to build.[6] A TP monitor is a software component that provides functions required for TP applications, such as transaction coordination, display management, and performance improvements.

NTT selected the ACMS proposal as the base of the new multivendor standard for two reasons: the ACMS TP monitor included a high-level TP control language called the Task Definition Language (TDL), which could be made portable more easily than a lower level API, and the monitor used a remote procedure call (RPC) communications model, which is easier to program than a peer-to-peer communications model. That is, the ACMS technology was determined to provide the best solution to NTT's requirements for multivendor portability and distributed processing.

The problems to be resolved by the consortium vendors, consistent with the principles of multivendorization set by NTT, were

- Portability
- Interoperability
- Common user access

Historically, portability has best been achieved among vendor platforms by using a high-level language such as C or COBOL. This principle was true for the MIA, except that the MIA consortium found it necessary to produce profiles of programming language standards. The C and COBOL standards are not sufficient to achieve portability because so many of the specification rules are subject to a variety of interpretations among vendors, and architectural language limits are not defined.[7,8]

An MIA profile of a programming language standard references the standard specification and modifies it to improve portability. In the case of the MIA COBOL profile, national text support is mandatory for portability of international language features. The X/Open Company adopted this work as the basis for their COBOL national language support and accordingly published the X/Open COBOL specification.[9]

The MIA COBOL profile also deletes sections of the ANSI COBOL specification that contain optional syntax that a vendor may choose to implement. Finally, the MIA COBOL profile sets common language limits such as the maximum length of a text string and the number of parameters supported on a procedure call. The resulting profile allows programmers to create source programs that are portable to any vendor who conforms to the MIA specifications.

The MIA programming language profiles were required because of the way vendor-driven standards are typically written. The goal of vendor-driven specifications work is to allow the widest possible interpretation of architecturally significant issues such as integer precision, file system naming rules, and memory manipulation, and thereby to allow the widest possible implementation and adoption.

The MIA C profile adds rules for defining the conversion of a signed integer into an integer of smaller or equal size and for defining the results of dividing by a negative integer. Neither of these semantics is defined in the ANSI specification because they tend to vary according to vendor architecture. The MIA C profile also defines wide-character handling in the print and file manipulation functions so that programs supporting international language character sets would be portable.

Efforts to address these portability issues, such as the X/Open XPG portability specifications, usually describe or catalogue the problems so that the programmer can avoid them.[10] MIA places the burden of ensuring application source code portability on the vendor instead of on the programmer.

No language standard existed for the MIA processing area of TP, however. Although some protocols existed for various degrees of interoperability, none existed for complete distributed transaction coordination.

Solving the TP Problem

Perhaps the most significant aspect of the MIA effort is its approach to resolving problems associated with distributed TP. Typically, TP applications are very large and involve strict requirements for performance and availability. TP applications implement the daily operations of a business. Some of the better-known examples include travel reservation systems and automatic teller machines. The term "transaction" is derived from the term "business transaction," which means an exchange of goods or money between two individuals or businesses, or some combination thereof.

Transactions, when automated, take on additional properties because computer systems are subject to failure in ways that manual systems are not. Computer systems are electrical, and electrical failures can damage data storage media. Computer systems are networked, and communication failures can interrupt the completion of a business transaction such as a travel reservation that requires the participation of multiple computers at multiple sites.

A computer transaction uses logging to ensure that business data is captured reliably or not at all. Perhaps most important, a computer transaction ensures that business computer systems recover quickly from any type of failure and begin processing data again without manual intervention.

Because of the highly demanding nature of TP, vendor implementations of TP system software depend on the features of specific hardware and operating system architectures for the purposes of performance optimization and fast recovery. The mechanisms for accomplishing fast recovery are complex and difficult to implement on a multiple-user system. Although business data is shared, operations on the data must be isolated so that one operation does not overwrite the effects of another operation. When two simultaneous requests arrive to update the same bank account, for example, the ending balance may be incorrect if the two updates are not properly serialized. Such errors can occur unless transactions are used to isolate and serialize the updates. Failures of media or communications can result in inconsistent data.[11]

These difficulties and others have deterred standards bodies from addressing the area of TP. Consequently, the market is dominated by proprietary solutions. Users are liable to be locked in to a particular vendor and to have difficulty achieving the benefits of competition.

The MIA TP specifications were designed to address these problems and to counter the shortcomings of the traditional vendor-driven software standardization process. MIA TP eliminates vendor-specific differences by adding a high-level language layer on top of proprietary TP monitors and by adding a common protocol at the lower layers for interoperation.[11] The only restriction that MIA places on the underlying software or platform is that it must be sufficient for implementing the specified TP functionality. Otherwise, vendor and user investment in existing systems is preserved.

The MIA consortium based the MIA TP protocol standard on the International Standards Organization/Open Systems Interconnection (ISO/OSI) TP protocol, and on the Open Software Foundation's (OSF's) Distributed Computing Environment (DCE) RPC, both of which were newly released.[12] To balance the risk of adopting a new technology, the MIA consortium chose IBM's Systems Network Architecture (SNA) Logical Unit 6.2 (LU 6.2) as a short-term alternative solution.

The MIA transactional communication specification combined DCE RPC as the data transport and OSI TP for the two-phase commit protocol. The resulting protocol was called the Remote Task Invocation (RTI) protocol, which was subsequently adopted by X/Open as the basis of their TxRPC specification.[13,14] Figure 3 shows the resulting MIA TP model.

To solve the portability problem, the consortium began with Digital's proposal based on the ACMS TP monitor's TDL and developed a new Structured Task Definition Language (STDL), which is a modular, block-structured language very similar to TDL.[15] The consortium eliminated vendor-specific syntax, ensured that STDL's features met NTT's user requirements, and conducted implementation studies to verify that the new language could be implemented on top of each vendor's existing proprietary TP monitors.[16] Figure 4 illustrates the layering of the new MIA TP language on the MIA TP protocol.

Because the MIA was based on standards as much as possible, the MIA TP work also had to be largely based on standards. Therefore, the STDL specification was integrated with the standard languages C, COBOL, and SQL to provide complete, portable application functionality.[17] The consortium mapped the data types among the four languages and specified interlanguage call semantics.

STDL procedures can call and be called by C and COBOL procedures. STDL implements the TP-specific functionality that standard C and COBOL lack. Examples of this functionality are beginning and ending a transaction, handling transaction exceptions, automatically restarting transactions, and coordinating multiple transactional resource managers (i.e., databases, files, and queues) locally or across remote TP systems in a network.

Adopting STDL as a new language represented a practical way to add TP-specific functionality in a multivendor environment while allowing the C, COBOL, and SQL languages to be used as specified in international standards. This approach did, however, result in additional integration problems. It was necessary to ensure that STDL procedures worked with C and COBOL procedures as well as with SQL and within the entire TP environment, which encompassed a large part of a platform's capabilities. An additional

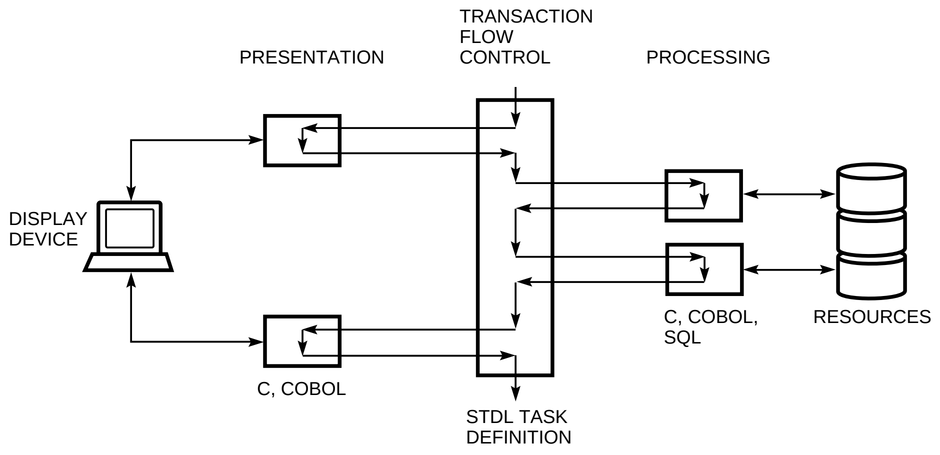


Figure 3
MIA Transaction Processing Model

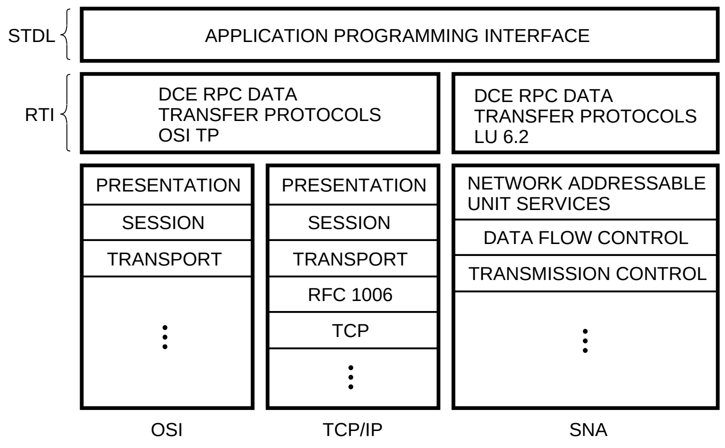

STDL
APPLICATION PROGRAMMING INTERFACE
RTI
DCE RPC DATA TRANSFER PROTOCOLS OSI TP
DCE RPC DATA TRANSFER PROTOCOLS LU 6.2
PRESENTATION
SESSION
TRANSPORT
PRESENTATION
SESSION
TRANSPORT
RFC 1006
TCP
NETWORK ADDRESSABLE UNIT SERVICES
DATA FLOW CONTROL
TRANSMISSION CONTROL
OSI
TCP/IP
SNA

Figure 4
MIA Transaction Processing API and Protocol

benefit results from the use of a compiler to check STDL syntax and semantics, thus reducing the instance of execution errors.

Implementing the MIA Specifications

Because the architecture was defined at the interface level, the implementation and system integration problem for vendors entailed identifying the components with conforming interfaces and assembling them on the platform that met the MIA specifications. Although focusing on three interfaces was practical with respect to completing the 11 volumes of the MIA specifications in approximately 18 months, such a scope left uncovered many areas of technology that the vendors intending to implement MIA would have to provide for themselves. System and network management, computer-aided software engineering (CASE) tools, and testing and debugging tools are examples of items that would have to be integrated with the components that complied with the specifications.

Table 1 lists the primary areas of the MIA specifications and the types of standards included in each area.[7,8,12,14,15,17–24]

The MIA specifications' practical approach to resolving the problems of portability and interoperability include carefully documenting where the vendor differences continued to exist among the implementations of the standards. "In general, the amount of information transferable between development and execution environments under the original MIA procurement specifications is less than that transferable when both environments are provided by the same vendor."[1] Some vendor-specific coding, for example, including file names in source code programs, could not be standardized by MIA because of fundamental vendor differences. Instances of such unresolvable problems were carefully documented.

The amount of portability gained by following the MIA specifications was significant, however, as compared to the amount that would be gained without using the specifications. The following example of defining the integer size illustrates the benefit derived from having the MIA C specification.

A C program written using a vendor's compiler that interprets a long integer data type as having 16 bits will not work correctly when ported to another vendor's compiler that interprets the same data type as having 32 bits (which is an acceptable interpretation according to the ANSI/ISO C specification). Typical solutions to this problem have been to document the problem and instruct programmers to recode when porting their programs, or to have programmers write their original programs so as to avoid the problem.

The MIA C specification resolved this problem and similar problems in that it represents agreement among the MIA consortium vendors on a common interpretation of the ANSI/ISO C specification. Because the MIA specifications are procurement specifications, vendors must conform to the MIA C specification when responding to MIA-compliant requests for procurement (RFPs) from NTT.

Table 1
Areas of MIA Specifications and Associated Standards

Areas of MIA Specifications	Standards
API	
COBOL	ISO 1989:1985, ANSI X3.23-1985
FORTRAN	ISO/IEC 1539-1991, ANSI X3.198-1992
C	ANSI/ISO 9899
STDL	MIA specification adopted by SPIRIT and submitted to X/Open
SQL	ISO 9075-1:1992
HUI	
OSF/Motif	OSF/Motif Style Guide, Release 1.2
IBM's Common User Access	No standard established
OPEN LOOK	No standard established
SII	
MIA TP protocol	MIA RTI specification adopted by X/Open as the TxRPC specification
OSI TP	ISO/IEC 10026-1:1992
MHS X.400	ISO/IEC 10021-1:1990, CCITT X.400-89
FTAM	ISO 8571-1:1988
TCP/IP, FTP, SMTP, TELNET, SNMP, UDP, CMIP	Internet protocol suite
X.25	ISO/IEC 8208:1990, CCITT X.25-89
ISDN	CCITT I Series
Ethernet	ISO/IEC 8802-3:1993, IEEE 802.3-93

Implications for Systems Integration and Application Delivery

NTT awarded Digital the first contract to deliver an MIA-compliant application. NTT selected its List Maintenance System (LMS), the application that manages the telephone number database used to produce telephone directories for all of Japan.[2] One purpose of the LMS was to sufficiently test the specifications. The LMS procurement involved 60 software products from a variety of Digital engineering groups. The components had to be modified to meet the specifications and then integrated, tested, characterized, and delivered on the OpenVMS operating system. The target configuration of three VAX 10000-630 systems in a VAXcluster configuration supported more than 10 client sites throughout Japan. The contract includes software, hardware, and services. Figure 5 illustrates the LMS application.

Of the 60 software components in the LMS platform delivery, 27 were required for conformance to the MIA specifications. Although the remaining 33 components addressed application areas outside the scope of the MIA specifications, these products had to be integrated with the MIA-compliant products, tested, characterized, and verified, thus making the integration effort more complicated.

Even though NTT realized some benefits from the standardized products that it procured according to the MIA specifications, it faced a dual systems integration problem. Delivery required complying with the specifications and also complying with the detailed terms of the specific RFP for the LMS.

Figure 6 illustrates the system verification and characterization process carried out by Digital's Systems Application Integration and Engineering (SAIE) group. This was the key effort in responding to the MIA-based procurement request.

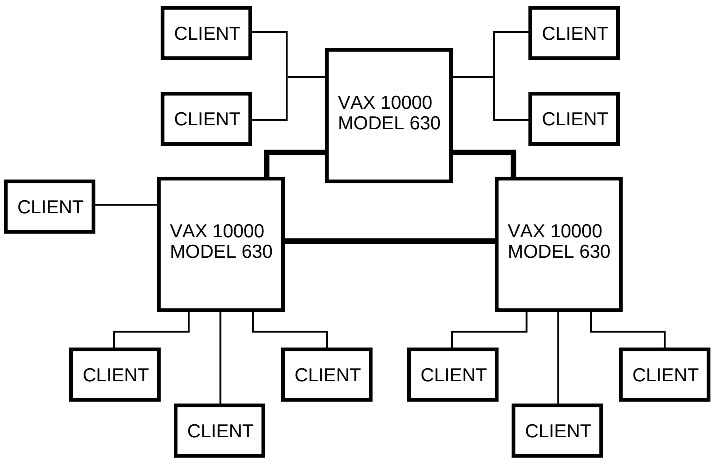
Digital established a special-purpose production systems program office (PSPO) to oversee the entire process of delivering the MIA-compliant RFP. This program office was modeled after the successful Alpha program office.[25]

A production systems board of directors represented the various engineering departments whose component products were included in the LMS. The board's function was to resolve priority and budget conflicts among the various departments. This group met monthly.

A special project forum was established with representatives of the individual products and engineers who could resolve technical problems and fix bugs that surfaced in the integration and testing activities. This group met weekly.

The SAIE group provided a "sandbox" for component product groups to install and test their products on the specific version of the OpenVMS operating system on which the components were to be delivered. This process was repeated for operating system upgrades and was made more difficult because initially a special version of the OpenVMS system was required to fully meet the terms of the RFP, in particular, to provide Japanese language support.

After the components were installed in the OpenVMS operating system, SAIE engineers verified that the components worked together by running test



SPECIFICATIONS:
3 VAX 10000-630 SYSTEMS IN A CLUSTER
11 CLIENT SITES
60 SOFTWARE COMPONENTS
STDL TP MONITOR
500-GB DATA REQUIREMENTS
MIA-COMPLIANT PLATFORM

Figure 5
List Maintenance System

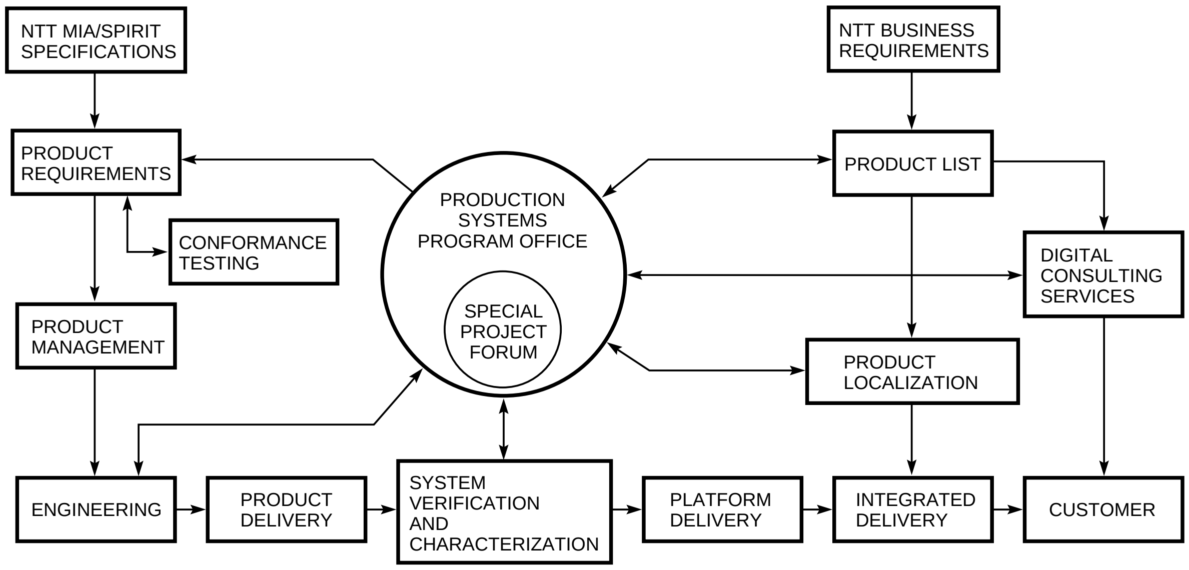


Figure 6
Digital's MIA Systems Integration Process

applications and characterized the overall performance of the platform as configured. Any problems that arose during this testing and characterization work were routed back to the component product groups by means of the special project forum. Finally, the program office coordinated the delivery to the local Digital office in Japan and to the customer (NTT).

The integration effort for the LMS uncovered more than 170 bugs, of which 25 were major obstacles. If Digital had not undertaken the integration effort, the problems would have shown up at the customer site and jeopardized the contract. Of the bugs, nearly 50 percent were directly related to integrating the various components on the common platform.

For example, one bug involved a fatal clash between versions of a threading package. Two LMS component products had incorporated incompatible versions of the same threading package without considering the potential problems that might arise if the two separately developed components were integrated and tested on the same platform.

Another problem resulted from the upgrade from the VAX C language compiler to the DEC C compiler, which was to comply with the new ANSI standard for the C language. While upgrading its C compiler to comply with the ANSI C standard, Digital altered the semantics of the associated run-time library. Most new software components are coded using C, so nearly every component on the platform was impacted.

During the 18-month period that the program office, the board of directors, and the project forum supported the LMS effort, 56 releases and patches were provided for LMS integrated products. Each time a new version of the operating system or a major component was released, the integration, testing, and characterization process had to be repeated.

The major lesson derived from the experience with MIA was the type of project and program management required to deliver a complete platform for enterprise-level computing on a large scale. Additionally, Digital engineers learned to work with other vendors to ensure the compatibility of Digital's implementation of the MIA specifications with the other vendors' implementations.

Digital remains very interested in pursuing opportunities to resolve enterprise-wide computing platforms for its large customers. The most significant problem to be solved is the systems integration problem. The MIA effort proves that products from different engineering groups within Digital need to be installed, tested, verified, and characterized before being delivered to the customer for use in a large application. Systems integrators can anticipate that the integration problems discovered during the LMS project will be compounded in an effort that involves software components from multiple vendors.

Large enterprise-level applications such as the LMS cannot be mass produced. The number of these large applications is small, and the needs of individual enterprises can vary significantly, even within a single industry segment such as telecommunications. Digital's experience with the SPIRIT consortium follow-on to MIA has demonstrated this.

It is therefore important to preserve the learnings about how the MIA platform was put together and, of lesser importance, to be able to exactly replicate the platform delivered to NTT for the LMS. Digital needs to be able to work with large customers such as NTT in the future and to complete large projects such as the LMS, backed by an internal systems integration and delivery organization.

Indeed, the systems integration problem grows more complex in a world in which products from multiple vendors are routinely required to work together in providing the solution to a large application's requirements. Customers tend to look more and more toward contracting for the technical expertise needed to solve these problems.

Delivery

Delivering an MIA-compliant business solution involves several levels of integration, each with its associated problems. The first level is integrating the required functionality in specifications developed by independent standards bodies. The next is combining standards-compliant component products on a single operating system and hardware platform, while preserving the required interfaces and behaviors. Third is incorporating the additional products and features necessary to develop a specific application on the standards-compliant platform. Fourth is ensuring that compliant platforms from multiple vendors can work together. The integrated product set must then pass conformance testing and verification. When application development begins, additional integration issues arise that affect the overall process.

During Digital's implementation of the MIA specifications and the subsequent integration activity to combine the components on one platform, several problems were discovered in the specifications. These problems were reported to NTT and directed to one of the specification working groups, which had continued under the auspices of the consortium for this purpose. For example, after testing interoperability using the RTI protocol, the mapping of communication errors to STDL exception codes was found to be incorrect.

Ultimately, not all the goals of the MIA initiative were met. During the implementation and delivery effort, it became apparent that specifying a standardized HUI would not be possible. The use of a windowing system with a common look and feel and common principles of operation (e.g., a mouse, icons, and pull-down menus) was sufficient for end users, and the industry players were too widely split to endorse a common solution. Specifying a standard for the size and shape of an icon or for how to entitle entries on a pull-down menu became unnecessary as windowing systems converged on common design principles of operation.

STDL Maintenance and Conformance

Because STDL was a newly specified language, it required considerable maintenance. NTT carefully monitored the vendor implementations of STDL to ensure that all the MIA vendors interpreted the specification in the same way. NTT procured several STDL-based applications from different vendors. Consequently, vendors were able to experience the inevitable implementation problems in realistic situations. If NTT determined that a problem was or might be related to the specification, it encouraged the vendor to submit a problem report to the appropriate MIA consortium working group.

NTT defined conformance testing for MIA, including STDL. Each vendor had to submit its completed platform for testing. Wherever possible, the MIA conformance tests were based on existing industry tests created by organizations such as the National Institute of Standards and Technology (NIST) and the X/Open Company. After passing each basic test, for example, proving conformance to ANSI C, a vendor had to pass an additional test for the "MIA delta," i.e., for the part of the specification that was different for MIA. In general, this difference consisted of Japanese language character support and more restrictive interpretations of a specification's optional or undefined parts. In the case of STDL, however, a wholly new suite of tests was needed to confirm conformance to the basic specification.

It became clear during this stage of the project that problems existed with the way in which the solutions had been specified. For example, the specifications for new TP technology had used existing standards specifications as models. In its eagerness to accomplish the task, the MIA consortium employed traditional methods of compromise and ambiguous wording to obtain agreement among the participating vendors. Not until the conformance tests began did the problem become apparent.

The conformance tests for STDL were divided into syntax verification tests and semantic tests. Conformance testing for any language is a tremendous undertaking because there are so many potential combinations of language syntax and semantics to take into account. The first problem for NTT was to reduce the number of tests to a practical amount, while keeping the results of the tests meaningful.

Initially, NTT took the approach of translating the specification's syntax rules into syntax tests and the general rules into semantic tests. The syntax tests were designed on the assumption that a vendor's STDL compiler would produce an error message for each violation of a syntax rule. The semantic tests assumed that a vendor's run-time system would produce an error message for each violation of a general rule. The specification had not been written using the same assumptions, however, and many of the syntax and general rules for the language elements contained a high degree of ambiguity concerning whether the rules had to be enforced at compile time or at run time.

Although this problem was never resolved for the STDL conformance tests, the tests were successful after they were redesigned to be more flexible in the method of catching errors. NTT was able to carefully monitor vendor implementations for consistency and compatibility.

MIA Applications

The intention of the MIA was to provide compliant software as the base, or heart, of a new application. MIA specifications standardize the most important interfaces and, consequently, enable users to realize the benefit of lower procurement costs, lower training costs, etc.

The MIA initiative was different from usual standards activities in that the implementations of the specifications were monitored by the same authority that caused the creation of the specifications in the first place. NTT bought systems based on its specifications, and worked with the vendors to maintain the specifications to correct problems that arose during implementation and application development.

For Digital, complying with the specifications meant implementing software to meet the terms and conditions of a large contract based on the specifications. Of course, the specifications covered only a portion of the overall platform and consequently did not address many conditions of the contract, such as CASE tools and system management.

Even though Digital's contract was for a single-vendor application, the source code had to be portable in case NTT decided to substitute another vendor's hardware for Digital's. Also, the new MIA-compliant LMS application had to fulfill at least the same functions as the old application. This application was therefore a good test of the MIA specifications; it would show how well the user requirements had actually been represented and met.

For Digital, the effort required delivering, for the first time, an integrated set of standards-compliant products for a large-scale business application. Digital had to combine components from a wide variety of internal product groups, make them all work together, and then upgrade or enhance the products to meet the MIA-specific requirements. In general, this entailed ensuring that our products were adapted to the Japanese market, i.e., that they supported the Japanese language character sets. In addition, the MIA required the integration of other new open technology, such as the RPC and other elements of OSF's DCE, DECmcc, and the new, ANSI-compliant version of DEC C.

Conclusions

Following the success of MIA, the MIA specifications became base input documents for the SPIRIT consortium, at which the user-driven standardization effort continues. Also input to SPIRIT were documents from AT&T, BT, Bellcore, and ETIS. The consortium model reduces vendor disagreements and yields a solution based on business requirements rather than on choice of vendor.

The fundamental requirement of the MIA was for a common computing platform for NTT's new enterprise applications that could be multisourced. This fundamental requirement is shared by the SPIRIT members, who represent the world's largest telecommunications corporations.

MIA and SPIRIT are seeking to lower costs in what has traditionally been the highest margin, lowest volume area of computing. The ultimate goal of a single, integrated platform that can be purchased off the shelf from a significant number of vendors does not appear to be completely attainable. Partial gains are more likely, as in the case in which suppliers integrate more or less dynamically the components of the required platform or platforms. Ultimately, the industry will be changed by the MIA and SPIRIT initiatives, although probably not in the exact way it was originally envisioned. For instance, since the MIA initiative began, the vertically integrated computer manufacturer, i.e., the manufacturer who supplies all the hardware and software components of the platform, has nearly vanished.

In the users' ideal vision, the software components conforming to the specifications in the MIA and SPIRIT platforms are off-the-shelf products that fit together easily. This goal has not proved to be the case in Digital's experience. Special product source code modifications were often required, and such modifications created integration challenges for Digital. For example, a special version of the DCE interface definition language (IDL) compiler was necessary to support the MIA. The new version mapped Kanji character set encoding to the ISO ASN.1/BER standard, whereas DCE RPC normally uses Numeric Data Representation (NDR) encoding.[26,27]

A paradox in the user-driven standardization effort derives from the fact that the MIA and SPIRIT platforms are intended for large projects, which are by definition limited in number. Therefore, creating off-the-shelf versions may be difficult due to limited platform volumes based on demand. For a vendor such as Digital, the effort appears to be best handled as a long-term partnership with large customers, supplying base technology and components to be integrated with those of other vendors. Integration becomes a continual and dynamic process. The key problem becomes systems integration, and a key question becomes who among the multiple vendors involved in supplying components will perform the integration.

The systems integration issue, therefore, is more important than ever before. As more and more vendors, pursuing their own core competencies, develop standards-based components, the greater the problem of component integration for customers who seek large-scale application solutions becomes. Enterprise-level platforms of the future are less likely to have components that are supplied entirely by a single vendor, and large applications, even standards-based applications, will continue to require platform customizations to meet the demanding requirements of these large users.

Acknowledgments

Thanks to Roger Baust, Bob Bell, Peter Conklin, Bo Erden, George Gajnak, Bob Howell, Bob West, and Steve Young for their help in delivering the MIA and for leaving a legacy of learnings.

References

1. *Multivendor Integration Architecture, Division 1, Overview, Technical Requirements* (Tokyo, Japan: Nippon Telegraph and Telephone Corporation, NTT Data Communications Systems Corporation, IBM Japan, Ltd., Digital Equipment Corporation Japan, NEC Corporation, Hitachi, Ltd., and Fujitsu Limited, 1991).
2. *Network Management Forum Proceedings, SPIRIT Tracks, General Meeting,* Marne La Valee, France (October 1994).
3. *SPIRIT Platform Blueprint,* SPIRIT 2.0, vol. 1 (Reading, U.K.: X/Open Company Ltd., Network Management Forum, 1994).
4. P. Conklin and E. Newcomer, "The Keys to the Information Highway," *Future of Software,* Chapter 3, D. Leebaert, ed. (Cambridge, Mass.:MIT Press, 1995).
5. *Multivendor Integration Architecture, Concepts and Design Philosophy* (Tokyo, Japan: Nippon Telegraph and Telephone and NTT Data Communications Systems Corporation, 1989).
6. R. Baafi, J. Carrie, W. Drury, and O. Wiesler, "ACMSxp Open Distributed Transaction Processing," *Digital Technical Journal,* vol. 7, no. 1 (1995): 34–42.
7. *Information Systems—Programming Language—C,* ANSI/ISO 9899 (Revision and redesignation of ANSI X.3159-1989) (New York: American National Standards Institute/International Organization for Standardization, 1989).
8. *Programming Languages—COBOL,* ISO 1989:1985 (Endorsement of ANSI X3.23-1985) (Geneva: International Organization for Standardization, 1985).
9. *X/Open CAE Specification,* C192 ISBN 1-872630-09-X (Reading, U.K.: X/Open Company Ltd., 1991).
10. *X/Open Portability Guide (XPG3),* ISBN 0-13-685868-6 (superseded by X/Open C, C214, ISBN 1-872630-39-1, COBOL dropped in latest version) (Reading, U.K.: X/Open Company Ltd., 1989).
11. J. Gray and A. Reuter, *Transaction Processing Concepts and Techniques* (San Mateo, Calif.: Morgan Kaufmann, 1993).
12. *Information Technology—Open Systems Interconnection—Distributed Transaction Processing—Part 1: OSI TP Model,* ISO/IEC 10026-1:1992 (Geneva: International Organization for Standardization/ International Electrotechnical Commission, 1992).
13. *Multivendor Integration Architecture,* Vol. 8, Div. 3, *Systems Interconnection Interface Specifications,* Part 4, *Remote Task Invocation Service Definition and Protocol Specification* (Tokyo, Japan: Nippon Telegraph and Telephone Corporation, 1991).
14. *X/Open Preliminary Specification, Distributed Transaction Processing: The TxRPC Specification* (Reading, U.K.: X/Open Company Ltd., 1993).
15. P. Bernstein, P. Gyllstrom, and T. Wimberg, "STDL—A Portable Language for Transaction Processing," *Proceedings of the Nineteenth International Conference on Very Large Databases,* Dublin, Ireland (1993).
16. E. Newcomer, "Pioneering Distributed Transaction Management," *Bulletin of the Technical Committee on Data Engineering,* vol. 17, no. 1 (New York: IEEE Computer Society, March 1994).
17. *Information Technology—Database Languages—SQL,* ISO/IEC 9075:1992 (Geneva: International Organization for Standardization/International Electrotechnical Commission, 1992).
18. *Information Technology—Programming Languages—FORTRAN—Extended,* ISO/IEC 1539: 1991 (Geneva: International Organization for Standardization/International Electrotechnical Commission, 1991) and ANSI X3.198-1992 (New York: American National Standards Institute, 1992).
19. *OSF/Motif Style Guide,* version 1.2 (Cambridge, Mass.: Open Software Foundation, 1992).
20. *Message Handling System and Service Overview—Data Communications Networks and Message Handling Systems,* Recommendation X.400-89 (Geneva: International Telecommunications Union, Comité Consultatif Internationale de Télégraphique et Téléphonique [CCITT], 1989).
21. *Information Processing Systems—Open Systems Interconnection—File Transfer, Access, and Management,* ISO 8571-1:1988 (Geneva: International Organization for Standardization, 1988).
22. *Interface between Data Terminal Equipment and Data Circuit-terminating Equipment for Terminals*

Operating in the Packet Mode and Connected to Public Data Networks by Dedicated Circuits—Data Communication Networks: Services and Facilities, Interfaces, Recommendation X.25-89 (Geneva: International Telecommunications Union, Comité Consultatif Internationale de Télégraphique et Téléphonique [CCITT], 1989).

23. *ISDN, I-Series Recommendations* (Geneva: International Telecommunications Union, Comité Consultatif Internationale de Télégraphique et Téléphonique [CCITT], 1989).

24. *Information Technology—Local and Metropolitan Area Networks—Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications,*ISO/IEC 8802-3:1993 (Geneva: International Organization for Standardization/International Electrotechnical Commission, 1993) and IEEE 802.3-93 (New York: The Institute of Electrical and Electronics Engineers, 1993).

25. P. Conklin, "Enrollment Management, Managing the Alpha AXP Program," *Digital Technical Journal,* vol. 4, no. 4 (Special Issue 1992): 193–205.

26. *Information Technology—Open Systems Interconnection—Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1),* ISO/IEC 8825:1990 (Geneva: International Organization for Standardization/International Electrotechnical Commission, 1990).

27. *Information Processing—Representation of Numerical Values in Character Strings for Information Interchange,* ISO 6093:1985 (Geneva: International Organization for Standardization, 1985).

Biography

Eric A. Newcomer
Eric Newcomer is a member of the Corporate Standards Group at Digital Equipment Corporation. As Digital's primary representative to the SPIRIT consortium in the United Kingdom and former representative to the MIA consortium in Japan, he works with representatives from other computer companies to create specifications for open systems software under the sponsorship of large information technology users. Eric joined Digital in 1984. He has 17 years of experience in database and transaction processing software. He holds a B.A. in American Studies from Antioch University.