M*Ware Dynamic Mediation Solutions

Managing, Integrating and Interconnecting Disparate Network Elements and Protocols

Introduction

Today's telecommunications industry landscape is evolving at a feverish pace fuelled by deregulation, competition, mergers, acquisitions and partnerships. Superimpose on these already complex dynamics, the many technological challenges created by the convergence of the traditional voice networks with the mushrooming data networks and the multiple enterprise networks each priding itself on being on the cutting edge of technology, and what you have, in vividly sharp focus, is a picture of an industry in the midst of a most major and traumatic transition.

All of this activity has created changes in the way the industry players work together:

- Carriers find themselves empowered to contractually obligate equipment manufacturers and software systems developers to provide specific interfaces for the systems they are purchasing;
- Network Equipment Vendors (NEVs) find that they need to be extremely flexible and have the ability to rapidly alter their design efforts based on the requirements of each of their customers;
- Operations Support Systems (OSSs) vendors find that they have to interface with both legacy and "tomorrow's" element management platforms;
- and Network Engineers find that they need to integrate a morass of multi-vendor network architectures that encompass a wide variety of both standard and proprietary interface protocols.

Emerging from these changes is a well-defined need for managing, integrating and interconnecting multiple and disparate network elements and protocols. Vertel, a long-time leader in developing and managing the interface protocols of the telecom and data communication industries, has focused its expertise to solve the growing problem of seamlessly mediating diverse network management interfaces.

Developed by engineers with years of experience working within the telecom environment, M*Ware Mediation facilitates the building of configurable and scalable mediation solutions for access, transmission, wireless, and switching networks along with a variety of operational support (billing, provisioning, service assurance, etc.) and database systems. These mediation solutions are designed to provide the following benefits:

- Rapid development time and greatly reduced time and cost-to-market due to the reusability of many components;
- A scalable, distributed architecture;
- Support for simultaneous multiple management protocols providing a robust solution capable of enduring multiple engagements;
- Ability to adapt to changes in the information model at run-time without coding and on-going development support
- Low system overhead thus preserving system performance specifications;
- Low cost of ownership through a thorough understanding of the mediation solution life cycle.

Vertel M*Ware Mediation

M*Ware Mediation is a full product line of pre-built and configurable components and solutions for the creation of a flexible and open network integration, data transformation, and communication management infrastructure.

M*Ware Mediation is made to seamlessly integrate existing networks and systems and to develop easily configurable off-the-shelf service applications required for end-to-end process automation.

This integration of management application and mediation functionality creates solutions where normal OSSs and integration code struggle and fail. M*Ware mediation components are the result of Vertel's experience in developing solutions for a variety of real network environments and components. This experience and the knowledge of a large number of management interfaces are reflected in the architectural design.

M*Ware Mediation aims at providing a mediation solution applicable throughout the lifecycle of a management system. It avoids the hard coding of interfaces or mediation semantics into the mediation code. This allows a customer to incorporate some changes to interfaces or mediation behavior without recompiling or even stopping the mediation device.

M*Ware carrier quality management development and integration greatly simplifies the task of developing and deploying mediation solutions. The distributed, fully component driven architecture is the result of Vertel's experience in designing mediation solutions for a wide variety of network environments and components. This experience and an in-depth knowledge of a large number of management interfaces and protocols are reflected in the robustness of the architectural design.

1.1 The different ways to implement Mediation

Conceptually, mediation solutions can be built in one of two ways: well-defined, one-time (hard-coded) solutions, or by using an approach which builds upon re-usable code elements and dynamically mapped data.

Hard-coded solutions tend to work in static situations – where both the interface and the network are not often subject to change. M*Ware high speed mediation addresses these customer needs.

However, in today's business environment that paradigm is becoming increasingly rare. In a rapidly changing environment, a more effective approach is a dynamic mediation solution designed from the onset with features that allow for life-cycle management.

As we illustrate later in this paper, mediation solutions need to have the ability to scale with the addition of new network elements, as well as changes to the protocol standards, OSS upgrades, etc., thereby protecting the investment made in the solution.

M*Ware dynamic Mediation accomplishes this life-cycle scalability by utilizing a data-driven approach governed by a configurable set of mapping rules. The meta-data approach and protocol normalization allows for a modular construct (saving both time and cost to market); while the mapping rules allow for changes in the interface at run-time without having to stop the application or re-boot the server.

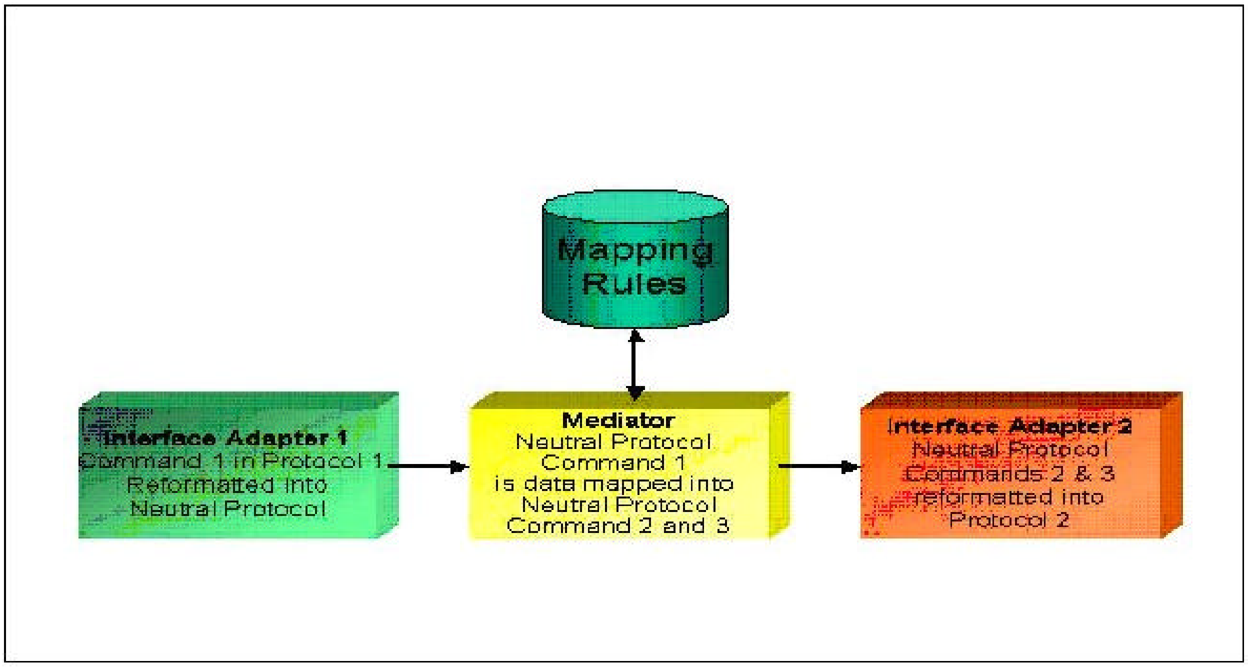


Figure 1: the neutral protocol and information model in the middle enables a component driven mediation that is far more re-usable, scalable and flexible.

1.2 The Mediation Process

In simple terms, the mediation process itself can be divided into four distinct steps:

- First, a management request is received according to a certain protocol (SNMP, CMISE, CORBA, ASCII, etc). This request arrives in a format compliant with the definition for that particular management protocol. The semantics of the request is translated into a normalized form.
- Second, the information models of the sending and receiving systems are mapped on the CIM, to define how logical mapping between the systems needs to be performed
- Thirdly, the normalized message is translated into one or more normalized outgoing management messages of what is most likely a very different management protocol. Predefined mapping rules are used to determine the nature and content of the outgoing messages to be generated.
- In the fourth step, the outgoing management request(s) are formatted and sent out according to the syntax definitions of the outgoing management protocol

The process is carried out in the reverse direction if a response to the request is received.

Functional Architecture

In order to accomplish this four-step mediation process, Vertel has developed a platform architecture that, through the use of re-usable interface modules and logical components (services and utilities) achieves engineering efficiency – directly translating into two critical customer benefits , i.e. rapid development time and reduced cost-to-market.

The following paragraphs will explain each of the M*Ware mediation components.

1.3 Protocol Neutral Layer

The information conveyed within network management messages is inherently similar regardless of the protocol language they are sent over. The real difference lies in the PDU (protocol data unit) formatting.

After much study, it was determined that most efficient way to perform the semantic analysis and data translation from one PDU format to another, was to represent the information in a protocol-neutral format.

The Protocol Neutral Layer was designed to this end. It consists of an extensible set of messages defined to represent the management PDUs of a variety of different management protocols. Translators for each of the PDU definitions in SNMP, CMISE, TL1, CORBA-IDL, along with some specific forms of ASCII, to their equivalent representations in the normalized format became a one-time effort and have been successfully accomplished.

(Using this “library” of translators, Vertel’s protocol engineers can help companies develop mediation solutions quickly and economically.)

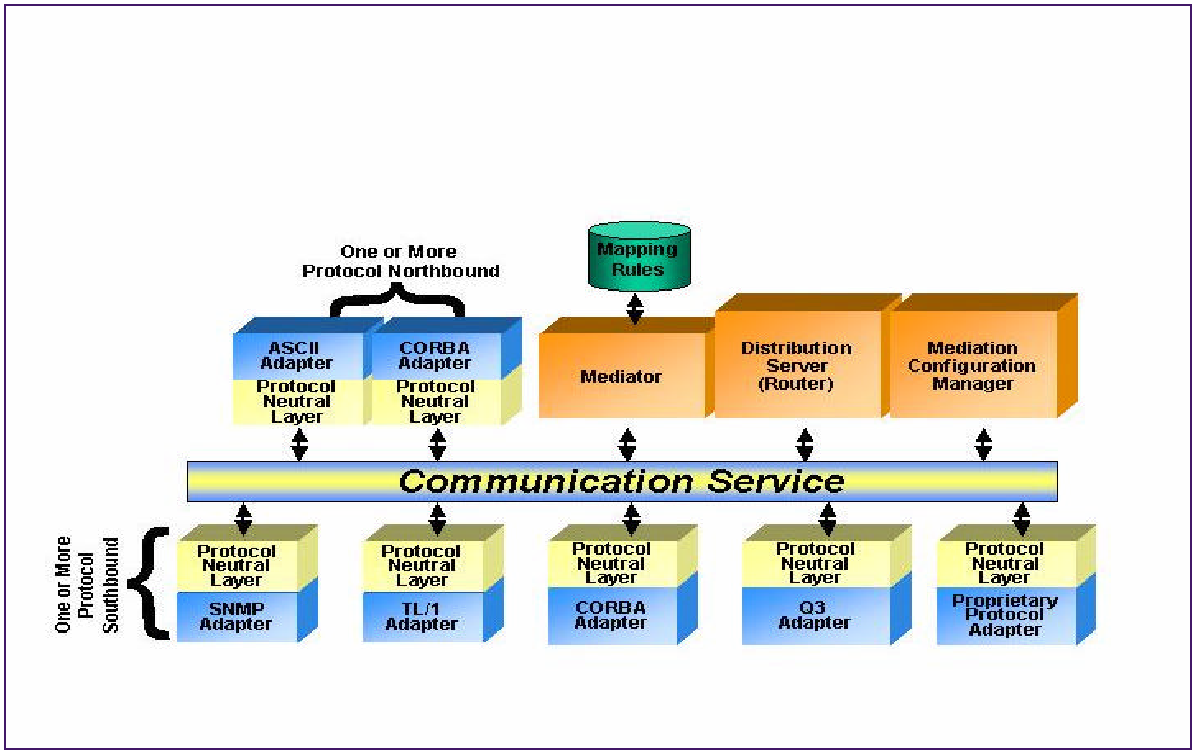


Figure 2. Functional Architecture

The message API exposed by this layer fosters re-use of most of the software developed for the interface adapters, while the informational model-specific formatting is performed through a generic meta-data APIs. All these APIs provide a framework for the easy application of a set of specific rules, allowing for the flexible specification of mediation mapping information.

The generic meta-data API allows support of dynamic MIB (Management Information Base) loading that enables the integration of multiple MIBs for different network elements or OSS interfaces. This API discovers the structure of the normalized messages via generic access functions. The PDUs are then formatted with the information from the meta-data. The resulting PDUs are subsequently sent out via the manager/agent binding modules (interface adapters).

The Protocol Neutral Layer therefore isolates the rest of the M*Ware mediation components (Interface Adapter, the Mediation module, the distribution service and the configuration manager) from changes in the information models and command sets of the various OSS application and EMS/NE interfaces.

Management Protocol Binding Modules (Interface Adapters)

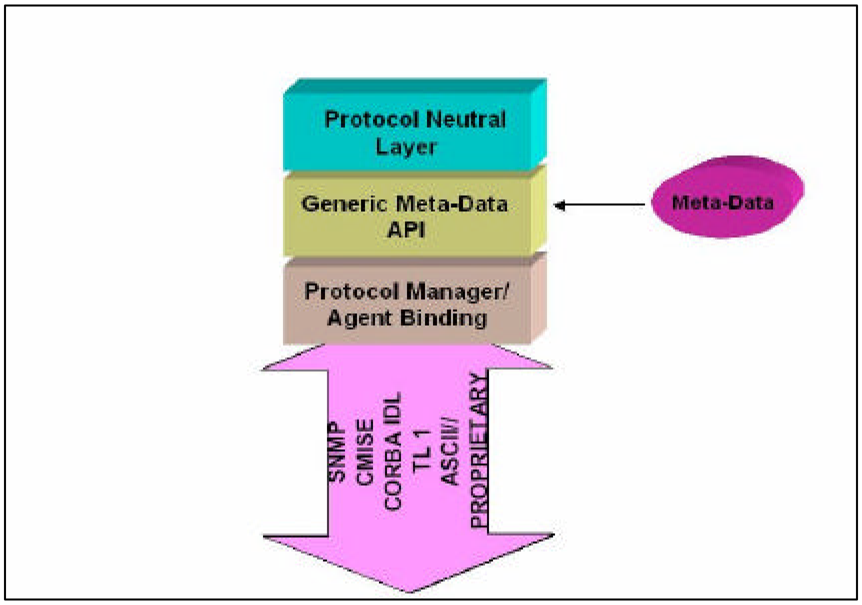


Figure 3. Interface Adapter Architecture

The Management Protocol Binding Modules, or more commonly known as the Interface Adapters, perform the function of formatting the management messages according to the different PDU definitions as defined by the specific management protocol. This module also provides the connection management function between the northbound and southbound systems.

The framework is designed to support CMISE, SNMP and MML (flavors of TL1, proprietary, etc.) interfaces to the EMS or NE directly. It also supports CMISE, MML (TL1, ASCII formatted, proprietary), CORBA IDL interfaces to various OSS applications. Support for the above mentioned protocols covers roughly 98% of the management interfaces found on today's installed base of network equipment, telecom OSS or data communication applications.

1.4 Software Recommendations

The following M*Ware products and shelf-ware are recommended for the integration of the various protocol binding modules.

- **Q3, SNMP, CORBA, ASCII, TL-1 or XML Interface to OSS Applications** – The M*Ware convergent Manager/Agent development Environment and associated tools is the agent development environment of choice for the development of an agent interface to communicate with an OSS application with any of these interfaces.
- **CORBA IDL Interface to OSS Applications** – Any industry standard ORB product (such as Iona Technologies' Orbix or Visigenic's Visibroker) can be used for building a CORBA server for implementing the IDL interface specified by the OSS application.
- **MML/ASCII/Proprietary Interface to OSS Applications** – There are currently no off-the-shelf products available for proprietary ASCII based management interfaces. Thus standard parser tools (e.g. lex, yacc, etc.) have to be used to parse the ASCII streams in order to interpret the information being conveyed by the command. Vertel Professional Services has, over the years, developed many ASCII parsers that can be very effective in parsing the more complex ASCII based messages.

All of the above, with the exception of the ASCII interface adapter, provide facilities for connection management either to the OSS application or the specific EMS/NE. The ASCII interface adapter can be modified to emulate the connection management characteristics of the OSS application / EMS application / or specific NE as required. Therefore connection management for ASCII interfaces, by its very nature, has to be custom developed.

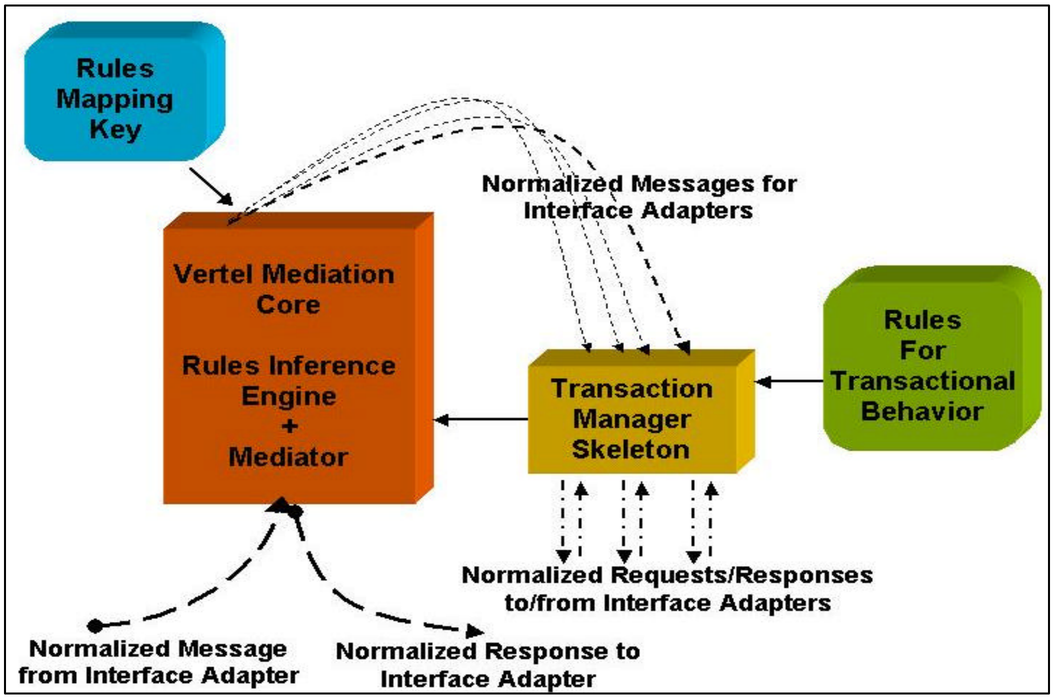

Rules Mapping Key
Normalized Messages for Interface Adapters
Vertel Mediation Core
Rules Inference Engine + Mediator
Rules For Transactional Behavior
Transaction Manager Skeleton
Normalized Requests/Responses to/from Interface Adapters
Normalized Message from Interface Adapter
Normalized Response to Interface Adapter

Figure 4. Mediator Module – Components and Interaction Diagram

The mediator module is at the core of the mediation process.
It is here that the actual step of mediating an incoming request into one or more outgoing requests is performed. The mediator module provides the structure for the analysis and generation of the normalized messages that deliver the required mediation behavior.

The mediator module uses a set of rules to perform analysis of an incoming request and spawn the new outgoing requests as dictated by those rules. Every message in the mediator module has already been normalized, thus facilitating the use of rules-based engine for data manipulation. The messages themselves are instantiations of a fixed number of message definitions modeling management requests from SNMP, CMISE, TL1 etc, and providing the user with a limited number of message objects from which to specify rules. This ability to represent many different management requests into a well-defined set of messages is a unique and valued feature of the M*Ware mediation platform. The OSS interface adapters and the EMS/NE interface adapters also interpret this well defined set of messages to communicate with the mediator module, thus facilitating the reuse of the adapter software for differing information models.

The Mediator Module can, itself, be broken down into several modules:

1.4.1 The Mediator Core

The Mediator Core Module is a rules-based application containing the rules inference engine. Additionally, it consists of a set of C++ classes that model the normalized messages. The C++ classes used to define these message sets are then rule-enabled by defining their structure within the rules language. Since the task of describing the classes that the rules manipulate is accomplished at compile time, it is very important that the number of messages that model management request PDUs from different management protocols be fixed. A great deal of time and effort has been put into defining a rich normalized message set to achieve the above.

The mediator core, as depicted in Figure 4, takes, as input, an incoming message. Appropriate rules within the rule set generate a suitable number of outgoing messages that are then sent to the appropriate interface adapters. The data manipulation intelligence is thus really performed via the rules, while the interface adapters perform the formatting of the mediated request/response.

1.4.2 Transaction Manager

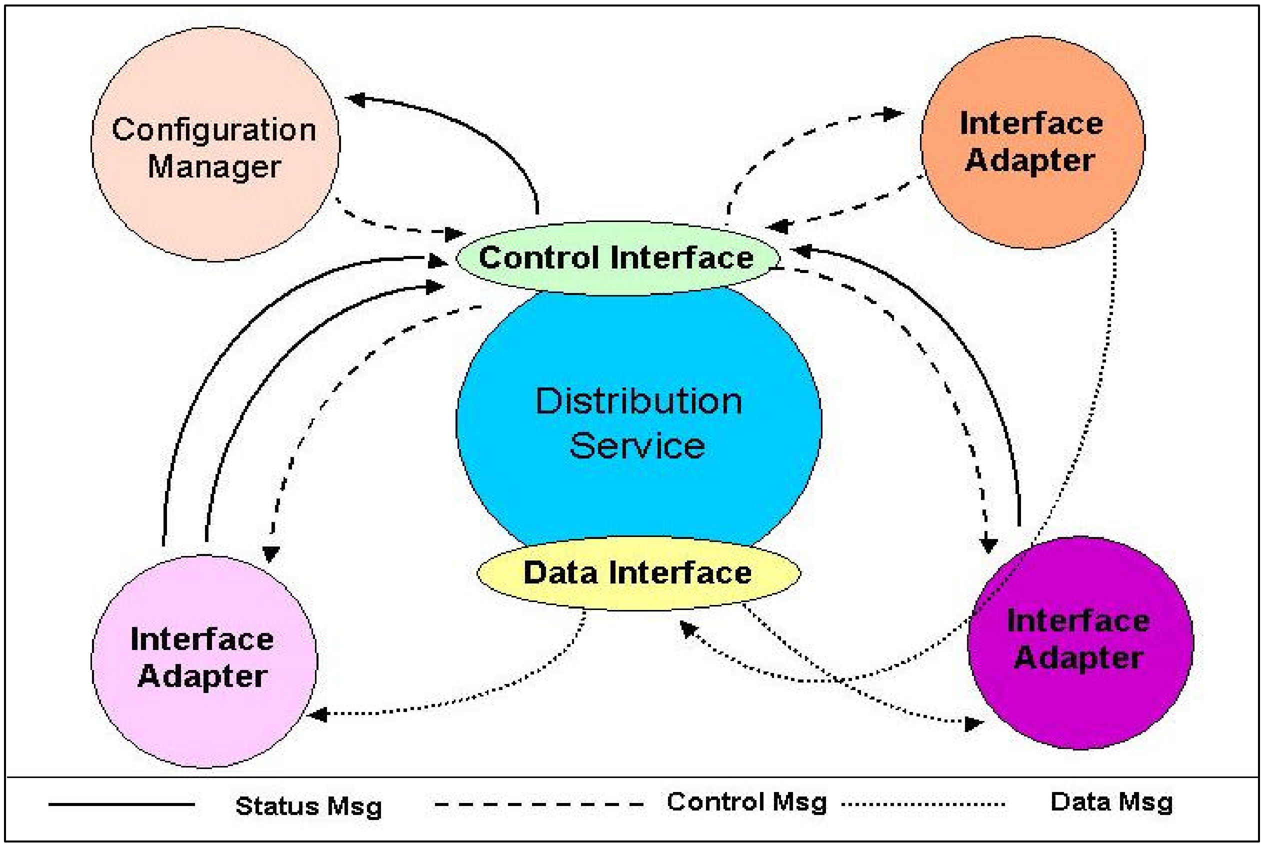
The mediator module also contains a transaction manager to correlate the responses of one or more outgoing messages to one incoming message. In light of the differences in transactional semantics of the different interfaces, the transaction manager in the mediator module is implemented as a configurable-state machine. The inputs to the transaction manager are the incoming message, the outgoing message(s) and a rule governing the transaction behavior which is to be implemented for each incoming and outgoing message set. The transaction manager is thus a powerful tool from which to implement different transactional semantics for different mediation requests.

The transaction manager core also utilizes the rule-based inference engine to provide an environment for the execution of rule-based transaction semantics. The transaction manager, just like the mediator core, provides a flexible architecture that emulates the different behaviors of the many network equipment management protocols in use today. The M*Ware mediation platform strives to achieve the flexibility necessary to adapt to different solutions wherever possible. This flexibility in the architecture comes from Vertel's extensive knowledge and experience building mediation solutions that cater to the complete mediation lifecycle of a deployed solution.

1.4.3 Mediation Mapping Keys

The mediation key is a set of rules specifying the mapping between two differing interfaces. One mapping key is needed to enable the mediation for each combination.

The rule-based language used in the M*Ware Dynamic Mediation platform specifies rules both at compile time and



run time and was specifically chosen for their run-time configurability. The ability to modify and add new rules at run time provides a very powerful mechanism to react to changes in OSS and EMS/NE management interfaces during version upgrades.

Hardwiring mediation logic in code is generally a dead-ended approach to building flexible mediation solutions. This methodology restricts the solution from being able to adapt to minor changes without an almost on-going commitment to development and testing. The rule-based approach, in addition to providing a flexible mechanism to adapt to, it provides a greatly reduced cost of ownership during the entire life-cycle of the mediation solution.

M*Ware dynamic mediation uses the C++ binding for rules engine. The rules are effectively written to manipulate the C++ representations of the normalized message sets. As mentioned earlier, the normalized message set contains a limited number of generic messages, which fully encapsulate the various details of different management PDUs.

The rules for the mapping keys thus only manipulate the C++ classes for the message set, allowing the user to be able to specify or modify rules within the framework. With a very short learning curve, the user can modify the mapping keys and extend the life of the mediation solution almost indefinitely.

The Mediation mapping keys for the systems are ASCII files containing rule sets written manually. Work is almost completed on a GUI front-end which will enable the user to be able to specify a rule quickly and easily.

1.4.4 Distribution Service (Router)

The Distribution Service module provides the routing function for the M*Ware mediation platform. The interface adapters and the mediator module do not interact directly, but rather via the distribution service. The routing of management requests from the OSS interface adapters and responses from the NE interface adapters (or OSS and OSS adapters) is deferred to the data translation phase in the mediator module. The mapping keys specify the routing for each of the management requests and responses in a flexible manner, allowing for plug-in and plug-out adapters and ease of changing routing information via rules.

The routing information is inserted into the normalized messages during the mapping phase by the mediator module and only interpreted by the distribution service to actually perform the routing function.

As illustrated in Figure 5, above, the distribution service operates on three types of messages: control messages; data messages; and status messages.

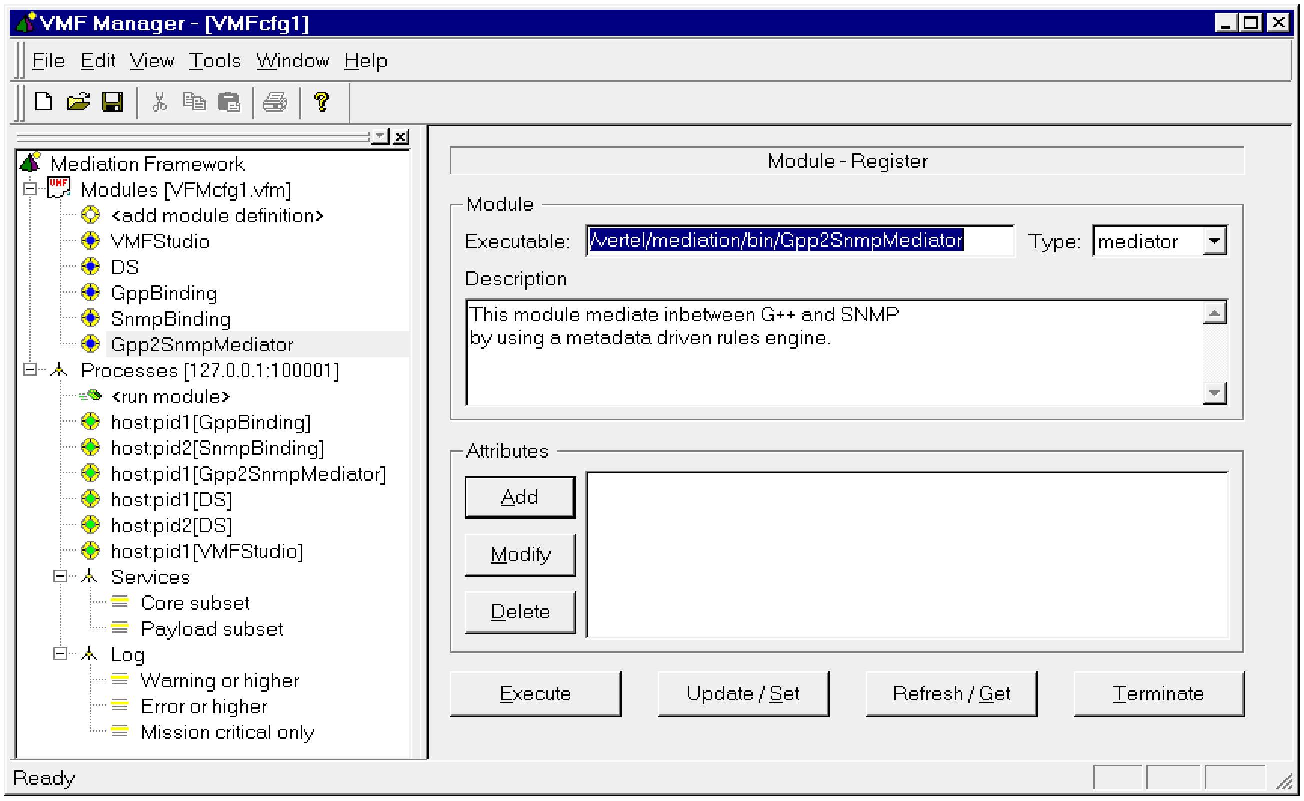
1.4.5 Control Messages

Control messages are used to convey configuration and status information to and from the interface adapters and the mediator module. Control messages for configuration of the interface adapters and the mediator module are sent out from the framework configuration manager to the distribution service and routed appropriately. The distribution service also creates instances of the interface adapters and the mediator module in response to the control messages. The distribution service is also capable of spawning instances of interface adapters and mediator modules on remote hosts using remote shell executions. The interface adapters and the mediator module also send out control messages to the distribution service reporting their health. The distribution thus also acts as a master controller of the mediation platform.

1.4.6 Data Messages

Data messages carry the actual management requests and responses to and from the interface adapters. The routing information in the above messages, unlike the control messages, is not inserted until the mapping phase to achieve flexibility in routing. The distribution service does not interpret the content of these messages, other than the routing information, and forwards the information to the appropriate module. The distribution service contains a database of routing information for each interface adapter and mediator module. This information is used to correlate the module ID with the physical address of the module to allow the distribution service to route the request. Like every other module the distribution service communicates with other entities using normalized messages.

1.4.7 Status Messages



Executable: /vertel/mediation/bin/Gpp2SnmpMediator Type: mediator

Description

This module mediate inbetween G++ and SNMP
by using a metadata driven rules engine.

Figure 6. M*Ware Mediation Configuration Manager Window

Status messages carry information about the current status of the framework. This is necessary, especially when multiple interface adapters are connected to the bus. The status messages provide timing data between the distribution service, mediator, and the various interface adapters.

1.4.8 Configuration Manager

The M*Ware configuration manager is a GUI tool for the configuration of the various components within the framework. The initial release of the configuration manager has been developed in the Microsoft Windows® environment, but a web-based tool is currently under development. The web-based interface will allow remote operation and configuration of the mediation solution, this enabling a means to even further reduce the overall cost of the solution.

Configuration for OSS Interface Adapters

- Host address and path of binary module for execution of this interface adapter
- Port address and connection management parameters for receiving connections from the OSSs
- Address information of the distribution service for communication of requests and status information

Configuration for Mediator Module

- Host address and path of binary module for execution of the mediator module
- Path information for various mapping keys to be loaded
- Address information of the distribution service for communication of requests and status information
- Path information for meta-data files to be loaded for the interpretation of VPN messages

Configuration for the NE Interface Adapters

- Host address and path of binary module for execution of this interface adapter
- Device list information to allow adapter to establish sessions with the devices to be managed
- Address information of the distribution service for communication of requests and status information

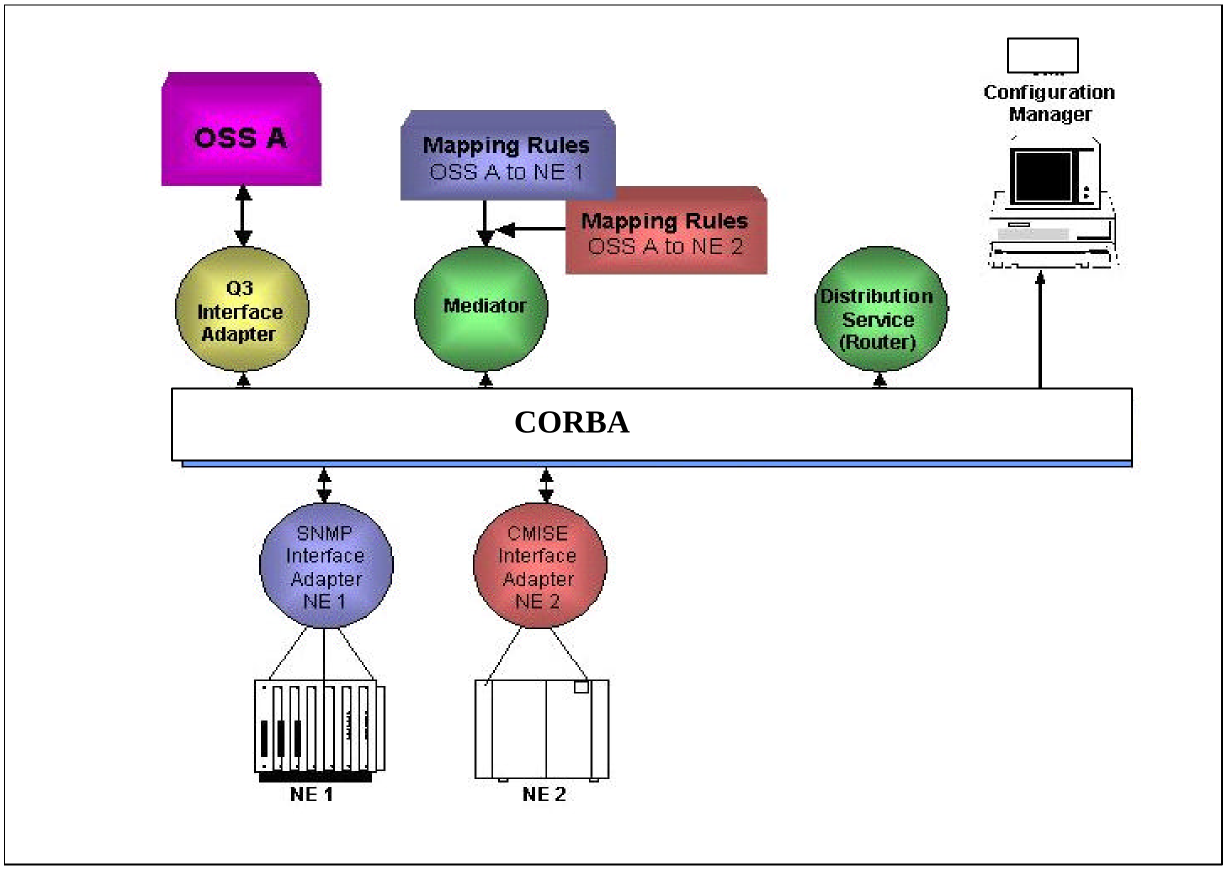

OSS A
Mapping Rules OSS A to NE 1
Mapping Rules OSS A to NE 2
Configuration Manager
Q3 Interface Adapter
Mediator
Distribution Service (Router)
CORBA
SNMP Interface Adapter NE 1
CMISE Interface Adapter NE 2
NE 1
NE 2

Figure 7. Initial Configuration: Mediation Solution for OSS A to NE 1 (SNMP) and NE 2 (CMIP)

The configuration information is entered via the configuration manager and communicated to the distribution service via VPN control messages. The distribution service is then responsible for starting the appropriate adapter and mediator modules and monitoring them. The configuration manager does not need to be running for the system to function. Once it is started it reads the system configuration from the distribution service to update its graphical display for the status and configuration of the different modules.

The configuration manager is thus a transient entity and is a time-sensitive view into the system configuration. The configuration and status information is actually stored and updated by the distribution service in its database. This architecture of the configuration manager facilitates its easy migration to a web-base tool that is stateless.

1.5 Mediation Life-Cycle

Mediation solutions are not inert and are susceptible to changes in the both the network architecture and the structure

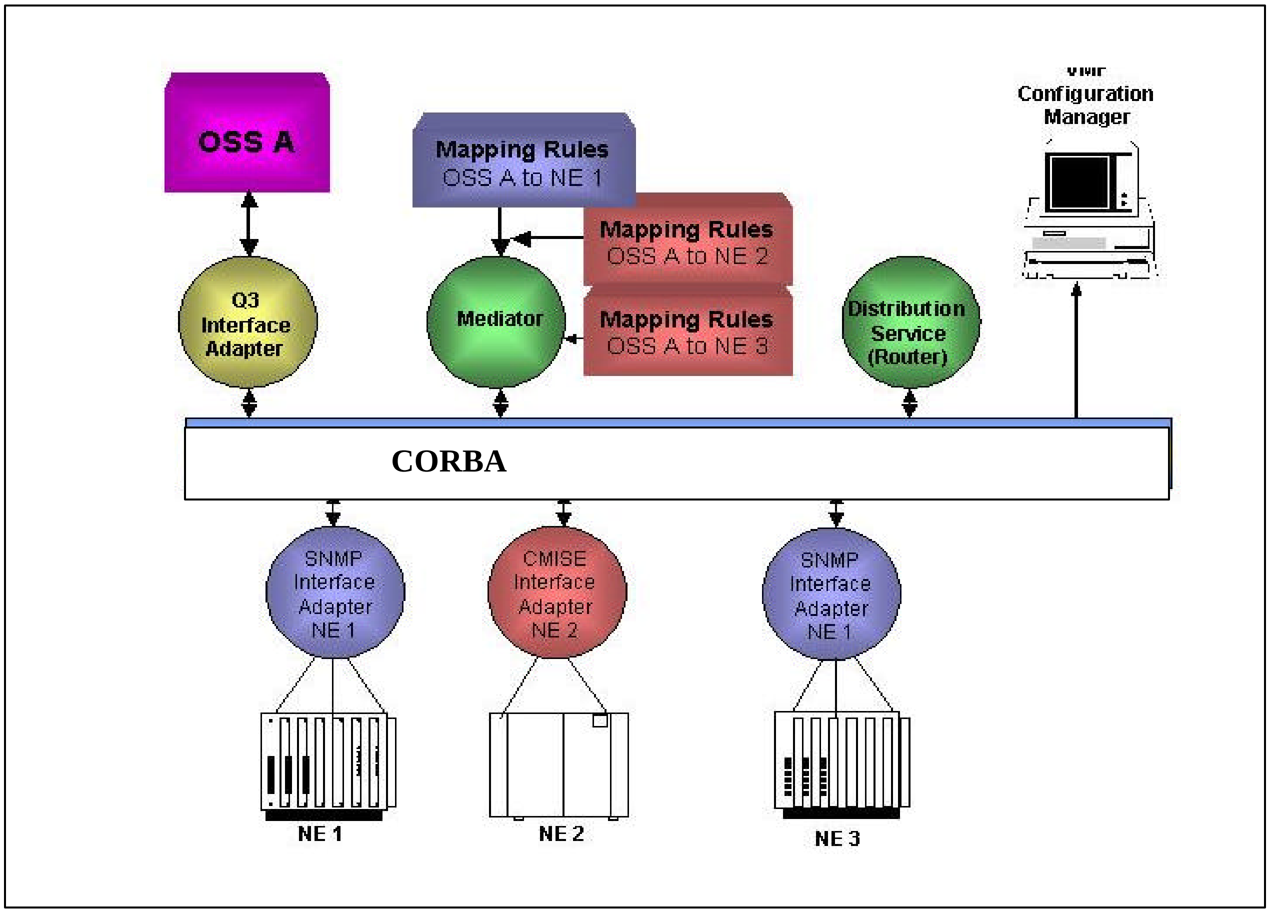

OSS A
Mapping Rules OSS A to NE 1
Mapping Rules OSS A to NE 2
Mapping Rules OSS A to NE 3
VMF Configuration Manager
Q3 Interface Adapter
Mediator
Distribution Service (Router)
CORBA
SNMP Interface Adapter NE 1
CMISE Interface Adapter NE 2
SNMP Interface Adapter NE 1
NE 1
NE 2
NE 3

Figure 8. Reuse of Interface Adapter for OSS A When NE 3 (SNMP) is Added. (Scenario 1)

of the varied OSS interfaces. It certainly not unusual for OSS and NE interfaces to go through feature upgrades after a "final" solution has been deployed. Network operators are always looking at ways to expand their networks to meet the ever-increasing demand for bandwidth – due to an increasing number of customers and their rabid hunger for new and improved services. This sets the stage for the carriers to engineer major overhauls in the network infrastructure in order to meet current and forecasted demand while remaining price competitive.

Due to the evolutionary nature of the telecom network it is important that any mediation solution infrastructure be capable of adapting to changes (minor and major) in the network architecture as well as software changes to the multiple OSSs and NE interfaces.

The mediation lifecycle is, therefore, the change process that can occur within a *deployed* mediation environment. The solutions built by Vertel, using its M*Ware mediation platform, enables the end-user to react effectively to maintain a viable mediation solution, even in a dynamic, evolving management environment.

1.6 A Sample Mediation Solution

Figure 7, above, illustrates a sample solution built around M*Ware Mediation. The main components of this solution include:

- Interface Adapter A for OSS A (Q3 Interface)
- Interface Adapter NE 1 (SNMP)
- Interface Adapter NE 2 (CMISE)
- Mediator Process
- Mapping Rules for OSS A to NE 1
- Mapping Rules for OSS A to NE 2

What follows in the next sections describe three working scenarios of the mediation solution life-cycle: the addition of a network element to an existing OSS; a modification of the OSS or Network Element interface; and third, support for a new OSS, OSS B.

1.6.1 Scenario 1 – New Network Element (NE 3) is Added – OSS (OSS A) is not Changed

Using the M*Ware dynamic mediation approach, it is very easy to add additional network elements, regardless of their interface protocol, into a pre-existing network architecture. To accomplish this integration, what is required is the design of the new interface adapter (which is again re-usable for additional engagements) and develop a new set of mapping rules (Mapping Rules OSS A to NE 3). This ease of integration is a very strong benefit of Vertel's framework approach.

1.6.2 Scenario 2 – OSS A or NE Interface Modification

It is not uncommon to see minor changes in the OSS and NE interfaces due to version upgrades. The solution built around M*Ware Mediation can adapt to these changes by incremental modifications to the rule based mapping information – without having to undertake any major development (and testing) programs. Certainly, as you may imagine, it may become necessary to modify the interface adapters, but only in the hopefully rare cases of major interface, or standards changes.

1.6.3 Scenario 3 – Support for OSS B

As illustrated in Figure 9., to add support for a new OSS, in this scenario, OSS B, a new OSS interface adapter needs to be implemented to adapt the OSS interface to VPN format. This is identical to adding support for a new network element! To support the mediation to NE 2 and NE n, appropriate mapping rules (OSS B to NE 2 and OSS B to NE n) need to be developed. The adaptation of the interface for OSS B to VPN also allows for the reuse of these adapters in on-going deployments of within the network, or networks of other customers.

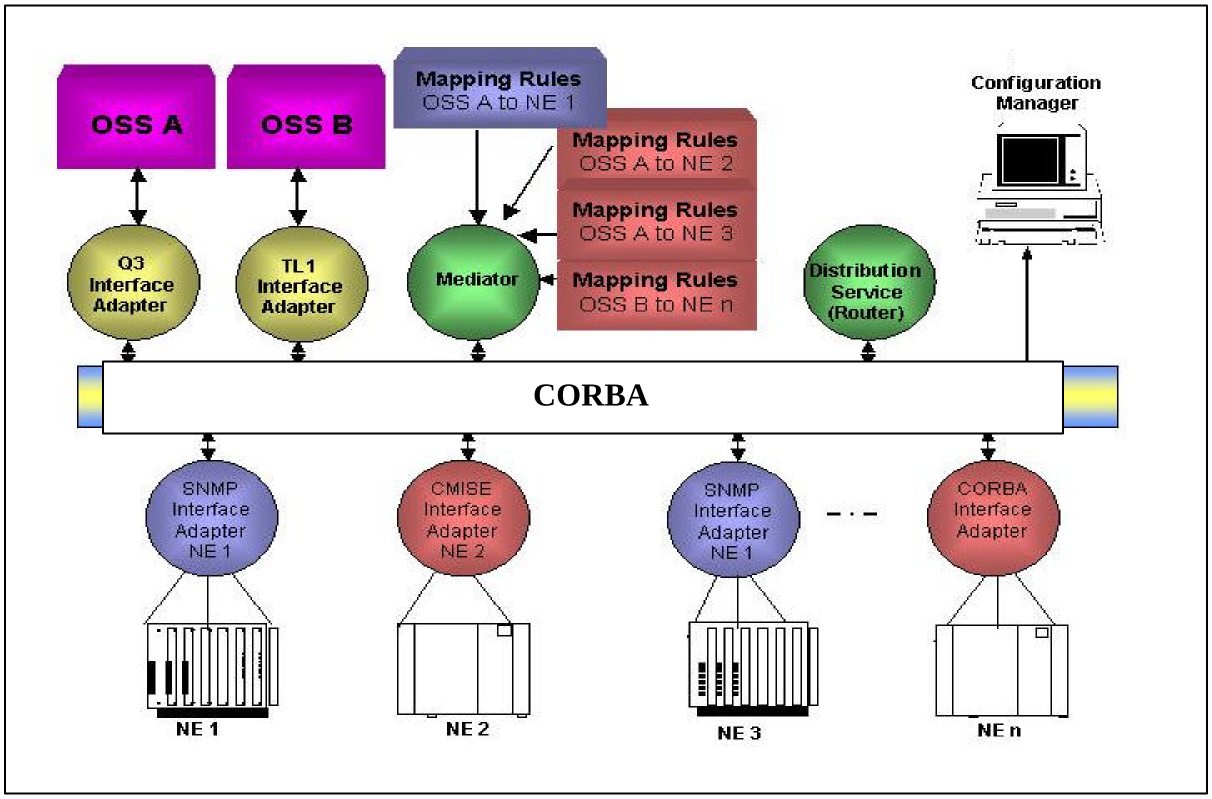

OSS A
OSS B
Mapping Rules OSS A to NE 1
Mapping Rules OSS A to NE 2
Mapping Rules OSS A to NE 3
Mapping Rules OSS B to NE n
Configuration Manager
Q3 Interface Adapter
TL1 Interface Adapter
Mediator
Distribution Service (Router)
CORBA
SNMP Interface Adapter NE 1
CMISE Interface Adapter NE 2
SNMP Interface Adapter NE 1
CORBA Interface Adapter
NE 1
NE 2
NE 3
NE n

Figure 9. Reuse of NE Interface Adapter for New OSS B

1.7 Other Features

1.7.1 Horizontal Scalability

A new mediator process can be started on a deployed solution to handle a portion of the OSS to NE combinations. The M*Ware architecture also allows for the support of multiple OSS and NE interface adapters on a single host. In order to improve performance, it is possible to partition the load over multiple hosts.

1.7.2 Vertical Scalability

A single mediator process can support the mapping rules for multiple OSS/NE mediation combinations providing for near unlimited vertical scalability.

Conclusions

Mediation solutions are fast becoming a critical path solution for network managers. Whether the need be created through mergers, acquisitions, business spin-offs, or a requirement to integrate a multi-vendor network that makes good business sense, there is a defined need, in today's telecom network, to have disparate systems communicate with each other.

M*Ware Mediation provides a cost efficient means to create these communications bridges. The use of a pre-defined and distributed component platform enables fast development time that directly translates to a faster time-to-market. With the ability to re-use components from one engagement to another means reduced costs and a better bottom line.

Additionally, the M*Ware distributed platform provides benefits that are required when one analyzes the overall life-cycle of a mediation solution:

- Scalable and distributed architecture
- Reusable components
- Support for multiple management protocols
- Ability to adapt the changes in information models without major development
- Ability to preserve performance requirements
- Low cost of ownership
- Ability to build mediation solution rapidly and with low cost

We would like the opportunity to discuss your mediation needs and how M*Ware Mediation can help your organization to speed-up time-to-market while greatly reducing your overall time-to-market.

For additional information, please call your regional Vertel Sales Executive, or call us at our Woodland Hills, Calif. headquarters.

About Vertel

Vertel is a leading provider of Mediation, Network Integration and Management and B2B Exchange Solutions.
Since 1995, Vertel has provided solutions to over 300 companies, including telecom infrastructure vendors, operators and service providers such as Alcatel, Nokia, Siemens, Motorola, Lucent, Nortel, NTT, Samsung, AT&T, BT, Deutsche Telekom, Cingular and Williams Communications.
Vertel's in-depth knowledge and commitment to industry standards, combined with experience of working with many different equipment types, allows the creation of high performance solutions that enable customers to quickly overcome technological barriers.
Vertel's mission is to make its customers successful by enabling them to reduce operational costs and introduce new services, networks and OSSs whilst maximizing existing investments.
Vertel's Professional Services organization in USA, Europe and Asia develops communications software solutions tailored to individual customer requirements and offers project management, systems analysis and other technical services.
For more information on Vertel or our M*Ware products, contact us at 21300 Victory Boulevard, Suite 700, Woodland Hills, Ca. 91367; telephone: + 1818 227 1400; fax: +1 818 598 0047 or visit www.vertel.com