

Architecture for integrated telecommunication management platforms

Telecommunication management network (TMN) systems represent diverse telecommunication domains with wide-ranging functionality which frequently employ different technologies and produce isolated solutions. Yet as telecommunications become more sophisticated, there is a growing need for telecommunication management integration and interoperability. The paper presents a general TMN computing-platform architecture which is flexible and powerful enough to support contrasting TMN management applications while facilitating integration. To validate the architecture, three diverse management applications were profiled based on components of the TMN platform architecture. Core infrastructure components, required by a broad range of TMN applications, are identified and examined and TMN-specific technology is highlighted.

by V. Wade, W. Donnelly, K. Riley, S. Roberts, D. Harkness, A. Carr, R. Shomaly and J. Celestino

1 Introduction and motivation

The last five years have seen a rapid growth in the capabilities and interoperability of general-purpose computing platforms. Much of the sophistication which was originally built into telecommunication management systems is now being offered by these emergent platforms. Because of the difficulty and expense of redeveloping computing platforms (CPs), the telecommunication management network (TMN) community must be able to utilise the technologies used by general-purpose CP systems and to adapt them to their specific needs. To achieve this saving in development cost and effort, TMN management systems have to be designed in such a way as to avail themselves of these CP systems. However, such general-purpose systems will never provide a complete solution for telecommunication management. The telecommunications community must therefore identify and incorporate TMN-specific technologies and develop these

within a TMN computing platform infrastructure.

2 Experience developing a TMN CP infrastructure

A general TMN CP architecture, capable of supporting a very wide variety of TMN management applications, has been developed by the authors over the last two years.^{1,2} This general TMN CP architecture was the result of the prototyping experience of several independent RACE projects, namely ADVANCE, AIM, NEMESYS and GUIDELINE, and represents a consensus of opinion. As well as this experience, the development of the general TMN CP architecture adheres to TMN related standards.^{3,4}

The general TMN computing platform for management applications consists of five layers. The goal of these layers is to abstract away from the complexity and heterogeneity of the underlying host systems and communication protocols. The layers also provide transparencies for distribution, information

access, and replication. These five layers are:

- (a) CP kernel
- (b) distributed processing support
- (c) computing platform interface
- (d) TMN support environment
- (e) user generic applications (Fig. 1).

The CP kernel consists of a number of components that mask heterogeneity and provides a common view of the world. This decreases the knowledge that an application programmer requires, and increases portability, interoperability and vendor independence. As it provides abstraction from both host operating systems and native databases, the CP kernel consists of a communication handler, transaction support module, storage handler (access to files and databases), threads manager and various device drivers for control of heterogeneous devices, e.g. printers etc.

The distributed processing support layer essentially provides abstraction from the distribution of the applications. This involves trading and binding to locate and gain access to remote resources, placement management which controls application instances and load balancing, transaction management (across the distributed hosts/databases), invocation and replication management.

The computing-platform interface provides the link between the programmes of the application developer and the underlying processing support. This link has off line and on line (run time) aspects. It does not necessarily encompass all the tools used to develop applications. It contains precompilers and compilers, runtime libraries and a type manager (which would facilitate co-operation of multilingual application implementations).

The TMN support environment provides the platform services that are required by the TMN management applications and the

Project summary

The research described in this paper formed a substantial part of the recommendations of the RACE 1 Project R1003, GUIDELINE. This project is charged with providing co-ordination for several other RACE 1 projects towards an implementation of telecommunication management networks (TMNs) for integrated broadband communication (IBC). It was concerned with developing a common approach to TMN architectures and providing recommendations on advanced information processing technologies appropriate for TMN implementation. The research performed by this project made use not only of RACE material but also of outputs from standards-making bodies such as ISO, CCITT, ISO/NMF, ANSI/T1M1 etc. where appropriate. Indeed, the RACE TMN

community has made inputs to the appropriate standards-making bodies.

Within the GUIDELINE project, a special interest group was formed with the specific task of developing an integrated, general TMN computing-platform architecture. To provide a synthesis of ideas and achieve a common viewpoint, several RACE TMN projects were asked to participate. These projects were drawn from differing TMN management domains, e.g. network and customer administration (ADVANCE Project), maintenance (AIM Project), traffic and quality of-service (NEMESYS Project). The result of this work has been the development of a flexible TMN computing-platform architecture based on sound concepts and practical prototyping.

user generic functions to fulfil their respective roles in managing the telecommunications network. As such, the services that the TMN support environment provides are specific to the requirements of telecommunications management. These include a directory service containing network management specific information and object manipulation/management which facilitates applications accessing TMN managed objects.

User generic functions perform key tasks which are required across a range of TMN applications. These include, but

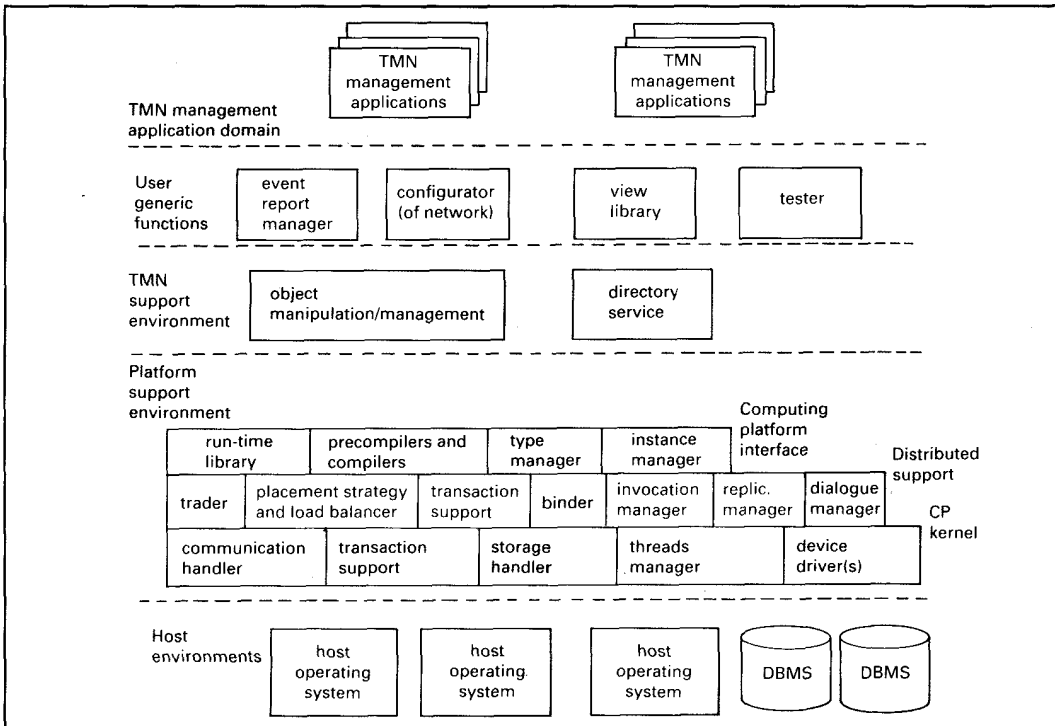
are not limited to, event report management, configuration (for the network) and view library (for abstracting and providing views of management information bases). This layer may also contain some ISO defined generic functions.

3 Three TMN management systems

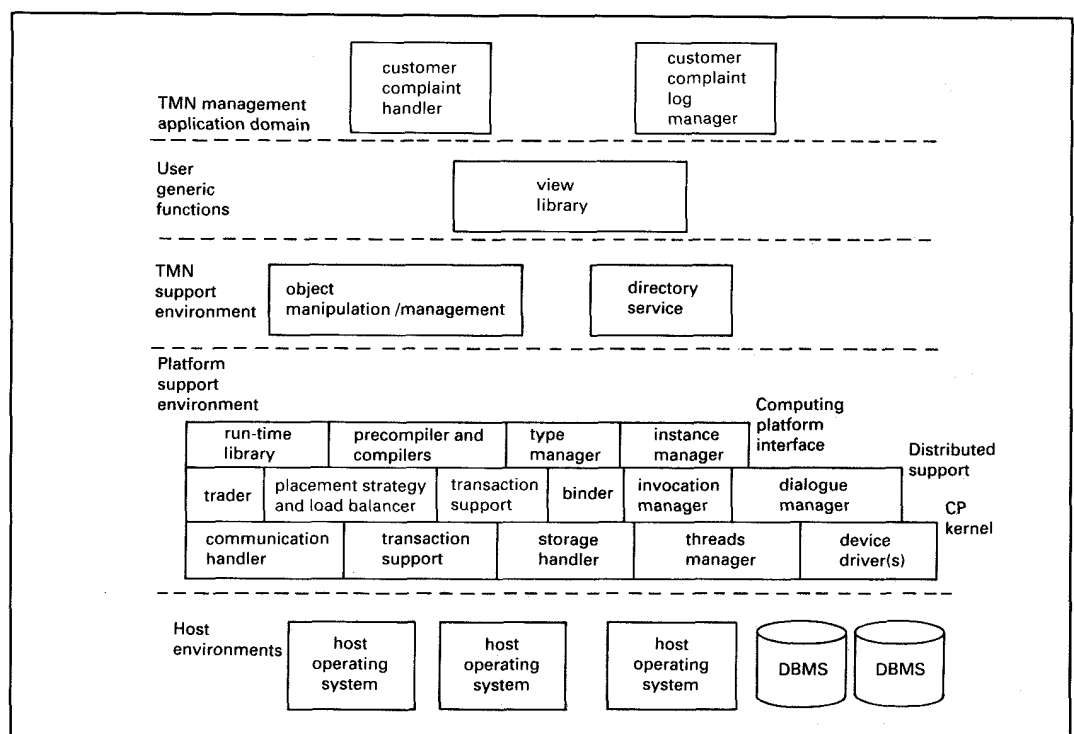
To test the general architecture, three management applications from different functional areas of the TMN were chosen.⁵ These management applications had been devised and developed by three RACE research projects

prior to the specification of the general computing platform. Thus, by adequately supporting each of these applications, the general computing platform could validate its usefulness across TMN management domains. A functional profile was developed for each chosen management system. Each of these TMN management applications is presented with its corresponding platform profile.

Customer complaint handler management application (CCH)
The customer complaint handler



1 General TMN computing-platform architecture



2 Customer complaint handling platform profile

operates within the context of a customer/provider relationship. Complaints originate from the customer, who perceives unsatisfactory behaviour in some equipment or service which is being received from the provider (e.g. no dial tone when making a call). On the provider side, the provider seeks to resolve all complaints received in order to satisfy an adequate level of service. Thus the provider will seek to establish the reason for any complaints (i.e. correlate complaints with known fault reports, flag unresolved faults for further scrutiny by the fault repair service).

There are two major areas of interest, namely complaint transmission from the customer to the provider, and complaint processing/resolution by the provider. This system involves a customer who has management capability (complaints can be as a result of using a service as well as a result of managing a service). The customer has the ability to log complaints through his own complaint management application on the customer network management system. There exist two TMN domains, TMN 'A' being the provider management system and TMN 'B'

being the customer management system. Communication for the purpose of complaint logging between the two TMNs (across the X interface²) is achieved through a wide-area network.

Computing-platform profile for customer complaint handler

The customer complaint handler resides at the management application (MA) layer of the general computing-platform architecture.⁶ The user generic functions layer and the TMN support environment layer facilitate the high-level operational requirements of the customer complaint handler whereas the platform support environment layer facilitates the low-level computing requirement of the applications (Fig. 2). The following functional blocks are required by the CCH-MA within each of the computing support layers:

Management application layer: Management applications which play a role within customer complaint handling will be present at this layer:

- TMN-A (network provider system)
- Customer complaint handling management application
- Customer complaint logging management

- TMN-B (customer system)
- Customer complaint agent
- Complaint-fault correlation management
- Operator complaint handling management

User generic functions layer: The view library provides the CCH-MA with host-independent access to the platform. A specific requirement for this application is access to management information. These information stores may be distributed and multivendor. Examples on information bases include objects within a common information model shared between the two TMNs, private information kept within various relational databases, and management applications.

TMN support environment: Directory service: The common information model provides a unified and logical representation of the information objects and management functions of the system. However, the actual resources of the system are widely distributed and heterogeneous. To achieve location transparency and to facilitate access transparency, logical naming is applied to them. Thus a directory service is needed to provide mappings of logical names to logical addresses for all

Authorized licensed use limited to: TRINITY COLLEGE DUBLIN. Downloaded on January 15, 2009 at 12:53 from IEEE Xplore. Restrictions apply.

the functions and applications of the system.

Object manipulation/management: Many entities are represented and accessed in the form of objects by the customer complaint handler system. This functional block is needed to abstract away from the complexity of access and manipulation of these objects across the system.

Platform support environment: Computing-platform interface: The applications associated with the customer complaint handler are written in a variety of languages and development environments. The implementors of the system require various platform facilities such as:

- precompilers and compilers, which facilitate the reduction of run-time checking overheads
- run-time libraries, which enable various platform functionality to be added to the system
- instance management, which facilitates the creation and deletion of object instances within the system.

Distributed processing support: The support of this layer is required to provide transparency of distribution and access to information resources. The functionality of a trader and binder is needed to locate various system components and to establish logical communication channels across applications.⁷ Other functions, such as transaction management and placement management are used to enable creation of instances and provide co-ordinated access to information.

CP kernel: The customer complaint handler is required to run on a variety of host environments. This layer is needed to abstract away from the complexity of the underlying host components and to increase interoperability and vendor independence. The use of an object-oriented structure provides the application programmers with a common view of the system resources and facilitates easier access to functional components. The functions of the communication handler, processing and storage handler are required to provide common access to local and remote resources, to provide a uniform interface to processing and scheduling capabilities of the platform and to enable access to files and various native databases.

Generic maintenance system (GMS)

Maintenance consists of three primary functions: fault correlation, fault diagnosis and repair. These functions are implemented in a management application called the 'generic maintenance system' (GMS).⁸ Faults are detected by the delivery of event reports to the generic maintenance system from the managed objects representing the resources being managed. These event reports are correlated, resulting in a list of candidate faults. Diagnosis refines the suspected faults using deep model-based reasoning and testing of the suspect equipment. Finally, the repair of the equipment that has been established as actually being faulty is scheduled.

The generic maintenance systems are anticipated as being widely distributed throughout the network element layer of the TMN, each having responsibility for a local domain in the telecommunications network.⁹ These generic maintenance systems model the detailed functionality of the network elements that they manage. Links between domains are managed at the network layer of the TMN. Generic maintenance systems at this level model the connectivity of the domains managed at the network element layer. Thus at the network layer, the generic maintenance systems orchestrate the co-operation between GMSs at the network element layer. Given the frequency with which event reports will be delivered and the dependency of the services provided over the network on the reliable and timely operation of the network, maintenance is considered to be a time-critical application, particularly with regard to determining which equipment is faulty and its subsequent removal from service.

Computing-platform profile for generic maintenance system

This section describes the CP profile selected for the generic maintenance management application.

User generic functions layer: The user generic functions of interest to maintenance are listed below. Standards for the interfaces between management applications and these functions are currently being specified by the standards bodies.¹⁰

Event report manager: Maintenance relies heavily on the regular and reliable delivery of

event reports from the network for it to correctly model the current status of the network. The event report manager passes these messages to the applications for which they are relevant.

Configurator: Maintenance frequently needs to isolate equipment so as to perform tests without degrading the performance of the network and also bring repaired equipment back into service. Requests to change the status of equipment must be made via the configurator to avoid conflicts between applications.

Tester: Maintenance needs to perform tests on equipment to establish whether the diagnosed cause of a fault is correct and also to determine whether or not a repair has been successful. However, other management applications may also wish to perform tests (e.g. customer complaint handling needs to correlate complaints to actual faults). Each management application wishing to perform a test makes a request to the tester so that it can handle conflicting requests.

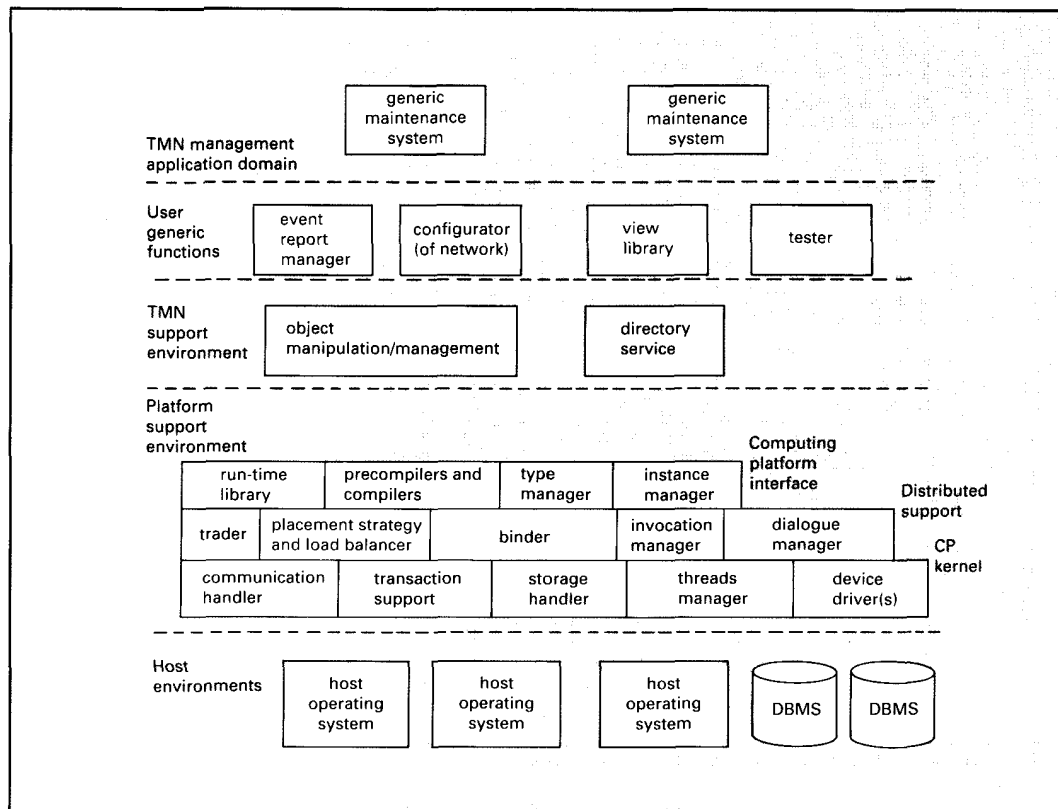
View library: Maintenance uses this function to access other components of the TMN and also to query databases providing persistency of the data used.

TMN support environment: Object manipulation management: When accessing common objects of the TMN, maintenance uses this function to avoid conflicts.

Directory service: Maintenance assumes that addressing between modules is by logical name and so expects location to be handled by this service.

Platform support environment: Computing-platform interface: The development of the maintenance system requires the use of all of these components, e.g. preprocessors, run-time libraries etc.

Distributed support: Maintenance has particular requirements on the trader and placement strategy/load balancer since it is necessary to state the preferred location of certain modules for them to perform adequately. Replication of generic maintenance systems within the same domain (e.g. to provide fault-tolerant processing) is not required, but automatic restart of modules executing on a host that has crashed is assumed. Binding is needed as well as the dialogue manager and invocation manager. Transaction processing is not used



3 TMN generic maintenance system profile

in the generic maintenance system but this will be needed for the smooth operation of the TMN.

CP kernel: Maintenance uses all of these features (with the exception of transaction support) but does not access them directly. Maintenance places particular demands on the communications handler. The generic maintenance system consists of a set of autonomous, co-operating agents. Co-operation between these agents is achieved by asynchronous message passing so as not to impose a fixed order on the maintenance cycle. Thus it is assumed that the communications handler is capable of supporting interactions of this type.

Fig. 3 illustrates the TMN computing profile for a generic maintenance system.

Traffic and quality-of-service management

The goal of this management application is to maximise the utilisation of network resources whilst maintaining an agreed level of quality of service (QoS) to users of the network. The approach taken is two-fold: firstly, to

determine call acceptance criteria at virtual path (VP) entry points in order to maximise virtual path bandwidth utilisation whilst maintaining a predetermined QoS; secondly, to maximise the proportion of successful connection attempts by optimising the bandwidth allocated to virtual paths and the routing of virtual paths. Both of these are addressed by the traffic manager, whereas the monitoring of QoS, generation of QoS warnings and provision of network load predictions for the traffic manager is undertaken by the service manager.

The primary components of the traffic manager are a call acceptance manager and a virtual path manager. The call acceptance manager adjusts the parameters of the network function which accepts calls onto virtual paths. The virtual path manager modifies virtual path bandwidth and routing-tables in the network.

Computing-platform profile for traffic and quality-of-service management

This section describes the profile, based on the CP-SIG

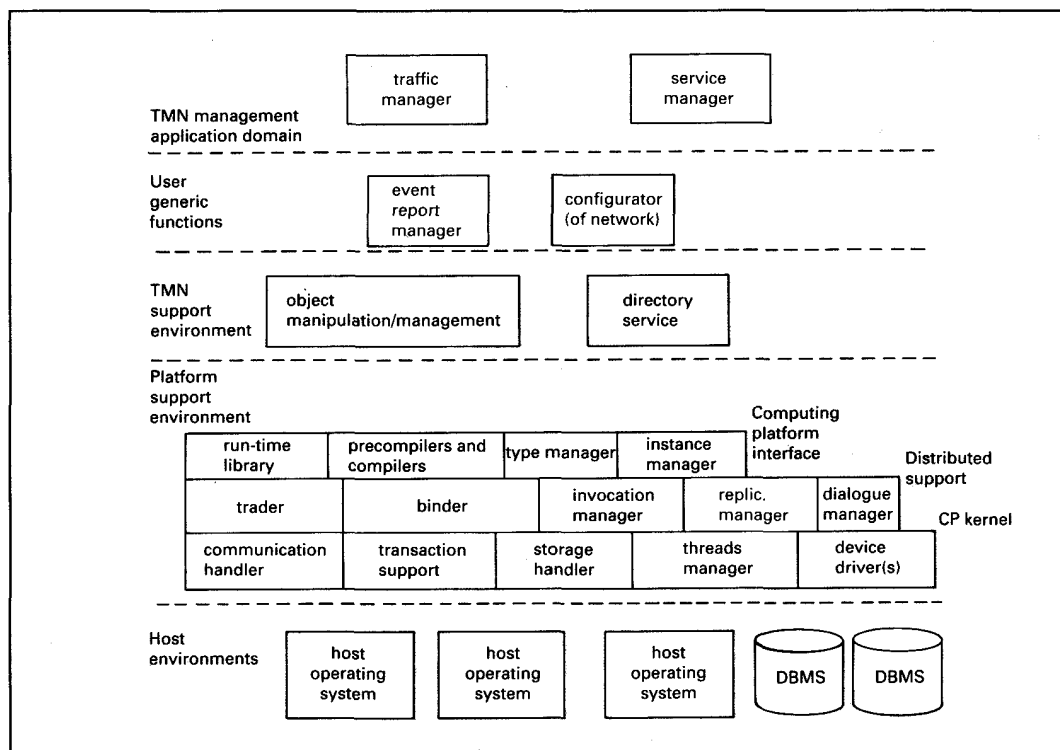
architecture, capable of supporting the traffic and QoS management system.

Management application layer: Both the traffic manager and the service manager¹¹ reside at the management application layer of the CP architecture.

User generic functions layer: Event report manager: The traffic manager receives virtual path events and network events, while the service manager receives events on service associations and QoS complaints. All of these are handled by the event report manager.

Configurator: The traffic manager interacts with the network in order to set routing and configuration parameters which effect virtual path bandwidth changes, and to set call acceptance function parameters. The service manager does not interact with the network directly and will therefore not make use of the configurator.

TMN support environment: Directory service: Traffic and service management are two distributed sets of applications which make use of location and



4 TMN platform profile for traffic and quality-of-service management

access transparency and therefore will use the directory service.

Object manipulation management: This provides access to an OSI-conformant management information base (MIB) as used in the service manager as well as private management information bases. For example, the traffic manager makes use of the management unit information base (MUIB) which provides a model of the network for performance management purposes.

Platform support environment: The traffic and QoS management system is a set of distributed applications designed to run a heterogeneous computing environment.

Computing-platform interface: All off-line and run-time support facilities of the computing-platform interface are used in the development of traffic and QoS management applications.

Distributed processing support: The application requires the use of location transparency. It therefore makes use of the services of the distributed processing support provided by the CP platform. Dialogue management and, more specifically, the user interaction management applications, identified in relation to the TMN

workstation function,² would be used in implementing the human interfaces.

CP kernel: The objects provided by the CP kernel which mask heterogeneity of the platform are of use to this system. Objects for transaction support will be of value in a multi-operator situation.

Fig. 4 illustrates the computing-platform profile associated with traffic and quality-of-service management.

4 Conclusions

The management applications were chosen to test the applicability and validity of the general TMN CP architecture. Each of these management applications was developed by different RACE TMN projects using divergent platforms and technologies. These systems reflected the specific requirements of the projects' domains and were not concerned with developments of a common interproject infrastructure. This paper has demonstrated that, by use of profiles, the general CP architecture is capable of supporting these diverse management systems. The paper has illustrated the necessary infrastructure from specific management systems down to

underlying host environments. This demonstrates the flexibility and applicability of the general CP architecture.

A second important result of the profiling work is that the selection of functional blocks of the general TMN CP architecture required to support each of the management applications shows a large degree of commonality. Thus the profiles provide explicit evidence for common infrastructure across TMN management systems. This suggests that a powerful means of achieving network management integration is via the general platforms' functionality. This means of management integration is significantly preferable to integration at the management application level.

Taking a broad view of the RACE community, it is envisaged that the TMN will be widely distributed and multivendor. Identification of the functional and nonfunctional requirements of the TMN CP is the first step in its realisation. The following requirements are commonly accepted by the RACE community:

- (a) The nonfunctional requirements include support for heterogeneous hosts, a high

level of performance, reliability, interoperability and fault tolerance. Extensibility is also an important nonfunctional requirement.

- (b) The functional requirements include storage, multilingual programming support and HCI support

The degree to which these various requirements are to be supported will depend on the type of applications that are using the platform. Maintenance applications operate at near-critical time scales and therefore a high level of performance is a priority whereas for non real-time applications, such as those related to accounting, reliability is a more important requirement.

Assessment of these functional and nonfunctional requirements in conjunction with the experience gained by the individual projects from their prototyping work allows for identification of the core technologies which must be part of any implemented platform for TMN.

Description of core technologies

This section gives a description, based on a consensus of the research projects, of some core technologies which exist at each of the layers of the general CP architecture.

User generic functions: In this layer the view library was identified as very important for inter and intra management application communications as it provides for mapping from languages and protocols used by the applications to the interaction language and protocols used by the rest of the systems. It provides a very necessary abstraction of information and functionality.

TMN support environment: A key component of any TMN installation is the ability to access and manipulate management objects. This object manipulation and management is therefore considered as core technology for a TMN computing platform. Another core technology is the directory service which is concerned with managing the name space of objects within the TMN.

Platform support environment: Because of the multivendor environment on which TMN operates, run-time libraries, precompilers and compilers are important services that the platform must provide.

The trader is considered essential to the general TMN CP

because of the distributed nature of the TMN and the requirement for distribution transparencies support (such as access and location transparency). The binder function, which provides a logical connection between two application entities, and the invocation manager, which is concerned with ensuring that interactions between application entities conform to the common computation model, are considered as core functionality of the platform. Because of its importance in providing access to the TMN by the user, some dialogue management functionality should be provided by the platform.

It is envisaged that the TMN platform will be supported by heterogeneous operating systems, databases and communication systems. Therefore the platform should incorporate communication handlers, storage handlers and device drivers to mask this heterogeneity.

Identification of TMN CP specific technologies

Although some core technologies have been identified for the general TMN CP, there still remains some *TMN-specific platform functionality* which is not expected to be supported by emergent general-purpose systems. A mapping of the platform support environment layer of the TMN platform architecture into several emergent distributed processing systems has been achieved,¹⁰ e.g. OSF DCE/DME, ANSA. However, the key TMN specific platform functionalities must be provided by the TMN community. The TMN-specific functionality tends to be concentrated at the user generic and TMN support layers of the general CP architecture. The functions identified as TMN specific include event report management, view library, object manipulation and management and directory services (X500) with specific TMN management objects. As the ISO/CCITT TMN management protocols evolve, extra communication handling support may also need to be supported.

Acknowledgments

This paper is the result of the research of the TMN Computing Platform Special Interest Group (CP-SIG), which comprises representatives from four TMN RACE projects: GUIDELINE, ADVANCE, AIM, NEMESYS. The authors would like to acknowledge

the co-operation and support of the CEC RACE programme. The architecture and conclusions presented in this paper are based on a consensus of opinion and research of four RACE TMN technology projects and form part of the RACE GUIDELINE recommendations for TMN platforms.

References

- 1 WADE V., DONNELLY W., HARKNESS D., RILEY, K., SHOMALY, R., CELESTINO, J., and CHAPMAN, M.: 'A framework for TMN computing platforms'. Proceedings 5th RACE TMN Conference, London, November 1991
- 2 'An implementation architecture for the telecommunications management network for the CEC RACE programme'. Working Group 2, RACE Project (R1003) GUIDELINE deliverable, GUIDELINE Main Event 8, 1992
- 3 'Common management information protocol specification'. Open Systems Interconnection, ISO/IS 9595, July 1991
- 4 'Principles for a telecommunication management network — Version R5'. CCITT Recommendation M3010, temporary document no. 40 (Rev1) Geneva
- 5 'Telecommunications management specifications'. NETMAN (R1012) RACE Deliverable, 1991, document no. 24/BCM/RD2/DS/A/006/B2
- 6 'MA scenario functional requirements capture (complaint handling MA)'. ADVANCE RACE Deliverable, pp. 12–23 ADBC0187
- 7 'ADVANCE Draft, the NCAS architecture'. ADVANCE Project Deliverable, document no. 09/BTR/AIR/DS/B/028/B1, 1992
- 8 STAHL, B., and AZMOODEH, M.: 'GMS specification'. RACE Project (R1006) AIM Deliverable, 1991
- 9 RILEY, K.: 'Design of GMS infrastructure'. RACE Project (R1006) AIM Deliverable, 1991
- 10 'State management function'. CCITT X.731, ISO 10164-2
- 11 'Experiment 3 design'. RACE Project NEMESYS R1005, Deliverable 10, May 1992
- 12 WADE, V., DONNELLY, W., ROBERTS, S., HARKNESS, D., RILEY, K., CARR, A., CELESTINO, J., and SHOMALY, R.: 'Experience designing TMN computing platforms for contrasting TMN management applications' in SMITH, R., MAMDANI, E., and CALLAGHAN, J. (Eds.): 'The management of telecommunications networks' (Ellis Horwood, 1992)

© IEE: 1993

Received 18th September 1992

V. Wade is with the Department of Computer Science, School of Engineering, Trinity College, Dublin 2, Ireland. W. Donnelly is with Broadcom, Ireland. K. Riley is with UNIPRO, UK. S. Roberts and D. Harkness are with Roke Manor Research, UK. A. Carr is with Cray, UK. R. Shomaly is with BT, UK. J. Celestino is with UPMC Versailles, France.