Journal of Network and Systems Management, Vol. 5, No. 4, 1997

Realizing TMN-like Management Services in TINA

George Pavlou[1] and David Griffin[2]

Telecommunications Information Networking Architecture (TINA) provides an architecture based on distributed computing technologies to enable telecommunications networks to support the flexible introduction and operation of new advanced services and to manage both the services and the network in an integrated fashion. While the service operation and management aspects are well advanced, network management aspects are less well defined. Resource Configuration Management (RCM) is one of the most important management areas as it covers, among others, the management of static topology and dynamic connectivity resources; these are both fundamental to the operation of TINA services. In this paper we present first an analysis of RCM, which results in introducing a new domain that deals with the configuration of management resources, in addition to network, service and computing resources. We then present a generic model for configuration management computational entities; this separates specific task-oriented aspects from generic resource representations accessed in a flexible fashion. The generic computational interface and relevant methodology for representing and accessing resources are influenced from OSI/TMN design principles, but make use of the TINA ODP-based Distributed Processing Environment (DPE). Based on this generic model, we present an RCM system architecture that deals with network and management resources. Parts of the latter have been verified through a prototype implementation in the context of a real field trial.

KEY WORDS: Network management; resource management; configuration management; TINA; TMN; network resource architecture.

1. INTRODUCTION

The TINA (Telecommunications Information Networking Architecture) [1] initiative aims at providing a framework for all telecommunications software, encompassing components ranging from connection establishment through network and service management to service delivery and operation. TINA supports

---

[1] University College London, Department of Computer Science, Gower Street, London WC1E 6BT, United Kingdom. E-mail: G.Pavlou@cs.ucl.ac.uk

[2] University College London, Dept. of Electronic Engineering, Torrington Place, London WC1E 7JE United Kingdom. E-mail: D.Griffin@eleceng.ucl.ac.uk

strongly the concepts of ODP (Open Distributed Processing) [2] via its modeling techniques and viewpoints, and the use of a DPE (Distributed Processing Environment) for providing the generic facilities required by all software running in a distributed fashion. One of the challenges of the TINA work is to bring together existing and established telecommunications software architectures, technologies, techniques and methodologies, such as the Intelligent Network (IN) [3] and the Telecommunications Management Network (TMN) [4]. TINA proposes an integrated framework which will address service deployment and operation as well as service and network management.

In this paper we are concerned mainly with the definition, design and implementation of TMN-like functions within the TINA Management Architecture and in particular with the management of Network and Management resources. It is a fair criticism of TINA to state that its work on the Service Architecture is much more mature than that of the Management Architecture, and that the approach by TINA has been to assume that TMN functions and management services can be taken more-or-less en masse and incorporated into the TINA framework. This paper examines some of the issues behind this assumption through a case study covering Resource Configuration Management (RCM). We have taken initial RCM specifications from TINA together with the requirements of a real prototype and enhanced them to create a design suitable for implementation. A major result of this design process is a general model for RCM which can be applied to all RCM areas. The resulting implementation has been tested and integrated in a comprehensive TINA-compliant system and trials have been undertaken to validate the design concepts as well as the implementation itself.

Our approach to this work was to bring a number of years of experience of TMN design and implementation to aid the development of a design which complied with the TINA objectives and techniques, using as much as possible of the incomplete specifications already produced by TINA. One of the initial hurdles to jump was the lack of a coherent methodology for system design coming from the ODP work which had been adopted by TINA. We are enthusiastic about the use of tools such as ODP viewpoints (enterprise, information, computational, technology, and engineering) for documenting the design of a complex, distributed system but we see them as just that—a set of documentation tools. There is no clear way to go from one viewpoint to another, and the relationships between, for example, the information and computational views are not specified anywhere. Our experience from the TMN area has shown that well defined methodologies exist for specifying and designing management systems [5, 6], leading the designer through management service definition and functional decomposition and then recomposition to arrive at a computational view.

The remainder of this paper describes a high-level view of the architectural issues related to the RCM management service, and demonstrates how the relevant TINA specifications were enhanced to include Management Resource

Configuration Management (responsible for the management of *management resources*) in addition to Network Resource Configuration Management, and how TMN and OSI systems management methodologies and principles were applied to these functional areas to greatly simplify the design and implementation of the resource map—the heart of RCM systems.

2. THE TELECOMMUNICATIONS INFORMATION NETWORKING ARCHITECTURE

The TINA-C (Telecommunications Information Networking Architecture Consortium) is an international initiative formed by telecommunications operators/equipment suppliers and computer vendors. Its main objective is to provide an architecture based on distributed computing technologies to enable telecommunications networks to support the rapid and flexible introduction of new services and to provide the ability to manage both the services and the network in an integrated fashion.

One of the main motivations for the TINA initiative was the modernization of the Intelligent Network (IN). The IN concepts, architectures and specifications provide a means for designing, deploying, and operating telecommunications services based on the telephony call model. IN operation is based entirely on control plane functions, with protocol based interactions between the embedded control entities in the local switches and the centralized service logic and data. The IN techniques have been successful for implementing enhanced telephony services, but it is more difficult to introduce modern, advanced services such as multimedia, multi-party communications mechanisms to support applications such as joint document editing, interactive distance learning, etc. Services such as these require advanced session management and control. Strictly speaking, more complex session control *could* be provided through signaling mechanisms, protocol based interactions, and centralized service logic, but traditional telecommunications engineering solutions, such as IN, are not as flexible as software engineering approaches based on object orientation and distributed systems.

TINA adopts the Open Distributed Processing (ODP) framework for specifying a ubiquitous software platform for service logic, covering both service operation and service delivery. The idea is that instead of being limited by the IN architecture, the telephony call model and signaling protocols, new advanced services may be deployed directly on a Distributed Processing Environment (DPE) and may be designed and implemented according to object-oriented principles and distributed processing techniques. In this way, service design and implementation can be achieved in a more flexible manner through re-usable software components while application inter-operability is achieved through the services of the DPE. This is a revolutionary departure for the telecommunications industry and is characterized by a shift from protocol-based telecommunications engineering principles to software engineering techniques such as Application Programming Interfaces (APIs), and component interface specifications which are more closely related to the programming languages used to implement the service logic.

The TINA framework is decomposed into four architectures: Computing, Service, Network and Management [1]. This discussion introduced the first two architectures. The Service Architecture defines the concepts and principles for the analysis, design, deployment, re-use and operations of service related software components supporting current and future advanced applications. The Computing Architecture specifies the framework for deploying and operating the software components of the Service and Management architectures; it achieves interoperability, distribution and component re-use through the facilities of the DPE. TINA has chosen the OMG Common Object Request Broker Architecture (CORBA) [7] as the basis for the latter.

The Network Architecture provides concepts for modeling the underlying network which provides the basic communications services required by the Service Architecture. TINA has based its modeling approach on the international Recommendations of the ITU-T, drawing on the Generic Network Information Model of M.3100 [8] and the SDH information model of G.803 [9]. The resulting specification is the Network Resource Information Model [10] which abstracts the communications resources forming the network infrastructure in a technology independent model.

TINA's Management Architecture covers the principles and concepts for managing TINA systems and networks and draws heavily on the ITU's TMN architecture [4]. The TINA specifications in the configuration management area are the most developed, especially those for connection management [11–13]. An interesting observation of the TINA results in this area is that they do not distinguish between the control and management planes in the same way that traditional telecommunications architectures do. Because of this, connection management is included in the Management Architecture as part of configuration management, rather than being part of the control plane of the Network Architecture supported by signaling mechanisms. This is perhaps the starkest example of the paradigm shift from telecommunications to software engineering principles.[3]

This paper concentrates on the Management Architecture of TINA, focusing on configuration management and especially on the network topology configuration management aspects of resource configuration management. This is dealt with in more detail in the next section.

---

[3] However, there is an ongoing debate on whether DPE-based connection management services can perform as well as tightly engineered signaling mechanisms for time critical call set-up procedures.

3. RESOURCE CONFIGURATION MANAGEMENT

3.1. Overview

The TINA Management Architecture is decomposed according to the five OSI functional areas, Fault, Configuration, Accounting, Performance and Security management (FCAPS) [14]. Configuration Management is concerned with the configuration of resources in the four TINA architectures. As such, it is more specifically called Resource Configuration Management (RCM). It was thought initially that RCM should only manage the resources of the Network, Service and Computing architectures [15]. We believe, though, that there is scope and necessity for managing the resources of the management architecture itself, as we explain next.

Layering is an important concept in TINA. Based on the TMN layering principles [4], TINA has defined the *Service*, *Resource*, and *Element* layers [1, 16]. The concepts of layering and decomposition of the overall architecture are orthogonal: each of the architectures can be split into Service, Resource, and Element layers. The management architecture supports management services which should be seen as specializations of general telecommunications services; as such, they should conform to TINA principles. Because of this, the Management Architecture itself can be considered to be layered according to the three different layers, and therefore contains management resources i.e., the computational objects implementing and providing the management services. These resources need to be managed just like any other resource.

The dependency between the layering and decomposition concepts leads to the conclusion that Resource Configuration Management is applicable not only to network resources, but also to service, computing and *management* resources. The classification of resources addressed by RCM are depicted in Fig. 1. TINA specifications recognize the fact that within the Configuration Manage-

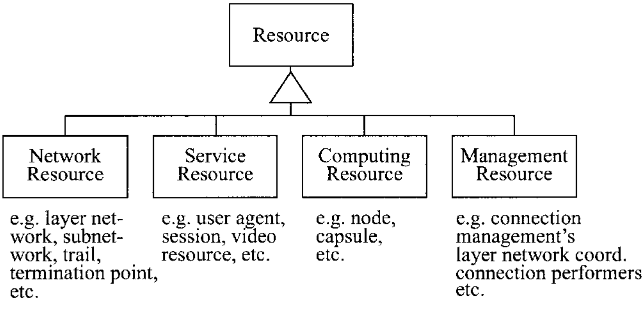

Fig. 1. Classification of resources.

ment domain, there exist Network, Service and Computing Configuration aspects but the specifications do not address Management Configuration aspects, at least not explicitly. Despite this architectural omission, there is clearly a role for Management Configuration Management in TINA as demonstrated through the following example.

According to TINA [17], Network Resource Configuration Management (NRCM) consists of Network Topology Configuration Management (NTCM), which deals with static network resources, and Connection Management (CM), which deals with dynamic network resources. The Connection Management part needs to be populated with computational components which are configured according to static topological information. In fact, Connection Management needs to be "managed" and this is done through the Connection Management Configurator (CMC) [11]. The latter is in fact a Resource Configuration Manager for Connection Management resources (Connection Performers, Layer Network Coordinators, etc.). In a similar way, although not yet defined by TINA, there could be resource managers for the other functional areas, e.g., fault, performance and accounting management.

This analysis leads to the conclusion that a new domain of Configuration Management is needed, namely Management Resource Configuration Management (MRCM), which we introduce to the TINA management architecture. The relevant set of managers are responsible for the "meta-management" of the TINA management architectural components. According to this view, it is now clear that the CMC belongs to the MRCM domain while it was previously thought (by TINA) to be part of Connection Management, and more recently as part of NTCM [17]. In summary, the configuration management functional area can be applied to resources in all the four TINA architectures, as depicted in Fig. 2.

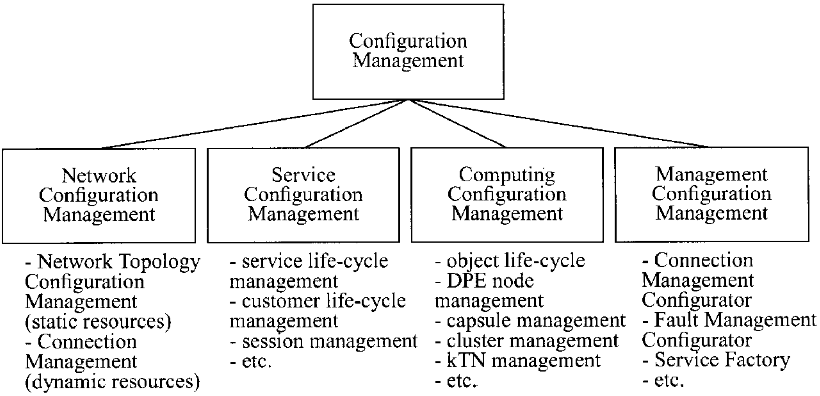

Fig. 2. Decomposition of configuration management.

3.2 Resource Configuration Management Functions

The requirements specified in [15, 17–19] and issues related to Resource Configuration Management in general, irrespective of whether it is Network, Service, Computing or Management RCM, can be summarized as follows:

- RCM should maintain a map of the resources under its influence in terms of two views: an inventory of all resources and a topology view showing relationships between resources.
- RCM should allow activation, deactivation, reservation and release of resources.
- RCM should support installation. Where physical installation is required, RCM shall emit a notification
- Self configuring resources should be supported by RCM. (This assumes that appropriate notifications are generated as resources are installed, configured or removed.)

Figure 3 is an informal representation of the scope of RCM. It deliberately does not map directly to an information, computation or engineering model. The figure is equally applicable to Network, Service, Computing and Management Resource Configuration Management. The circles contained in the resource map are repre-

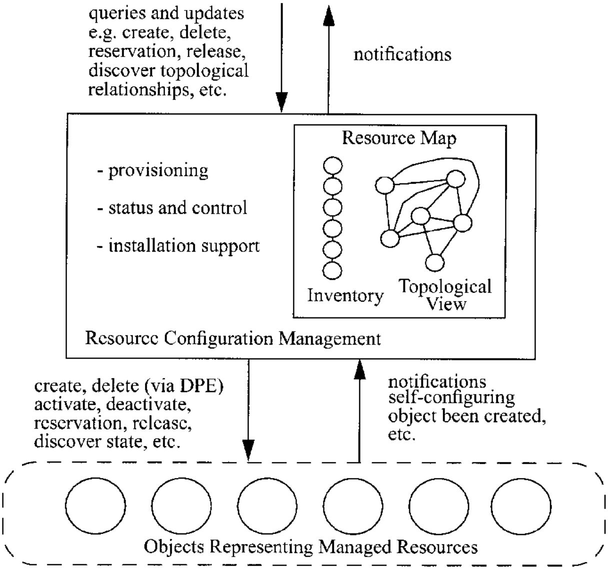

Fig. 3. The scope of RCM.

sentations of the managed resources which are separate objects outside of RCM. Examples of managed resources are:

- in the network architecture: subnetwork, termination point, etc.
- in the service architecture: User agent, Terminal agent, session, video/audio resource, etc.
- in the computing architecture: node, cluster, capsule, etc.
- in the management architecture: Connection performer, Layer network coordinator, Network RCM computational objects, Service factory, etc.

3.3. A Generic Model for Resource Configuration Management

A key aspect of RCM is the maintenance of resource related information, including relationships between resources. This information needs to be navigated in a flexible fashion, as required by query, update, discovery and inventory services. Since the resource information is object-oriented, a major requirement is an object-oriented database-like access mechanism.

There are two main approaches to providing interfaces to support such services. The first approach is to define specific operations, in what we term a *task-oriented* interface, tailored to the particular resources managed by the RCM component in question. The definition of these operations will depend on the individual requirements of the clients of the RCM component as well as on the specific configurable resources being managed. The second approach is to define a generic set of operations applicable to all managed resources and providing the basic query, inventory, etc. services required by any client component. The advantage of the first approach is that the operations may be seen as simpler by specific clients as they are tailored for their exclusive use. On the other hand a significant disadvantage is that new interfaces and operations need to be specified for every new type of client and resource.

The first of these two approaches is the one that seems most prevalent in the current TINA specifications, while the second approach is similar to that of TMN and OSI systems management. Our view is that both types of interface can co-exist, but the existence of the second approach is essential for generic Resource Configuration Management to allow re-use of specifications and software across all resource management areas.

A computational construct is required for this purpose, providing access to information objects specific to the nature of the resources and exhibiting behavior that maintains consistency with respect to resource updates. We use the term *Resource Configuration Map* (RCMap) to name such a generic computational construct.

The computational interface offered by RCMap is general, offering maximum expressive power, but on the other hand this genericity may not be always

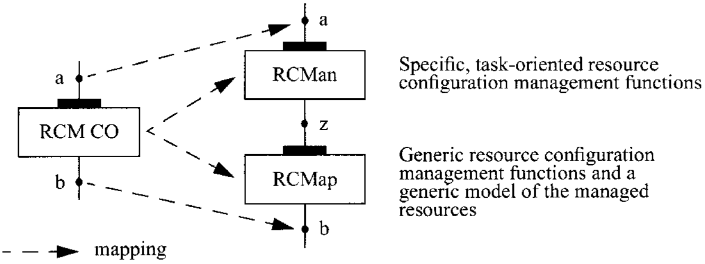

Fig. 4. Specific and generic RCM interfaces.

desirable and clients of RCMap may prefer simpler, *task-oriented* interfaces. These are provided by Resource Configuration Manager (RCMan) computational objects which act as clients of RCMap and provide specific computational query and update interfaces, tailored to the nature of the particular resources held in the map and to the requirements of a particular client or group of clients of RCM.

The RCMan/RCMap model is depicted in Fig. 4. This approach, where computational objects hold internal information objects and provide access to them in a task-specific fashion, is not uncommon in TINA. The session graph in the Service Architecture and dynamic resource information in Connection Management are typical examples. However in TINA the computational interfaces offered by the relevant computational objects are task-oriented rather than generic as defined earlier. This type of computational object, generically termed RCM CO, is depicted in the left part of Fig. 4. The separation of specific functionality from generic database-like access as exemplified by the RCMan/RCMap model is depicted in the right part of that figure. Note that the applicability of this model to existing computational specifications; e.g., in connection management, is evolutionary rather than revolutionary as it retains the existing specific access interfaces (interface a in the figure) through the RCMan part of the model.

Because of the hierarchical modeling principles used in TINA, computational objects similar may be layered hierarchically, for example the Connection Management Architecture and the recent layering of NTCM into EML, NML and layer network topology configurators [17]. In this case, relevant resource information is held in a distributed hierarchical fashion but there is no collective view of it through a single computational interface as is the case in Connection Management at present. When applying the RCMan/RCMap model to such hierarchical structures, there are two distinct possibilities, resulting in:

- a single RCMap, holding all the information previously held in a disjoint hierarchical fashion and accessed by hierarchically structured RCMan objects; or

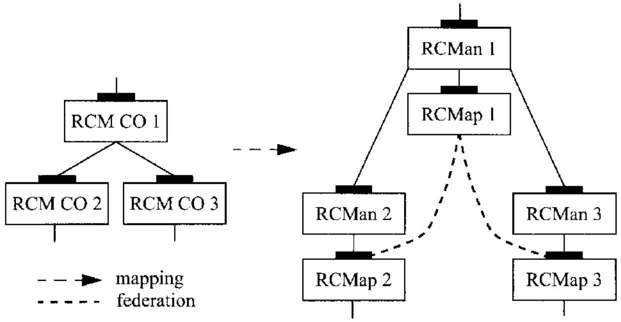

Fig. 5. A distributed resource map using federation.

- many hierarchical RCMan/RCMap pairs, with the topmost RCMap providing a global view of the overall resource space in a hierarchical federated fashion.

The second possibility is shown in Fig. 5. The choice between the two options depends mainly on scaleability issues since operations on managed resources should always be performed through the relevant representations in the resource map for maintaining consistency. Given the fact that OSI systems management-like principles are used for the computational interface of the resource map, a global federated view is feasible and similar approaches exist in today's OSI/TMN systems.

4. AN RCM SYSTEM COVERING NETWORK AND MANAGEMENT RESOURCES

Based on the generic RCM model presented in the previous section, we propose an architecture addressing the management of network and management resources, i.e., Network RCM (NRCM) and Management RCM (MRCM) domains. Network resources are further decomposed into static resources, representing topology information and covered by the Network Topology Configuration Management (NTCM) domain; and dynamic resources, representing connectivity information and covered by the Connection Management (CM) domain. The architectural decomposition of the NTCM and MRCM domains has not yet been addressed in TINA, in fact the MRCM is a completely new domain. On the other hand, CM is relatively mature [11] and we are reusing its architectural decomposition as currently specified.

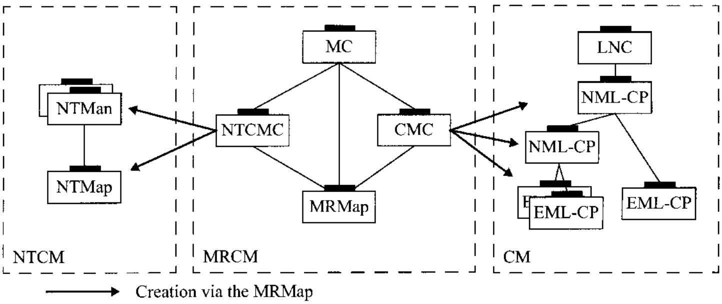

Fig. 6. General RCM architecture.

4.1. A General RCM Architecture

The overall RCM architecture covering the NTCM, CM and MRCM domains is shown in Fig. 6. The initial approach taken regarding both the MRCM and NTCM is that of a single resource map. The use of a centralized resource map represents only a first approach to its architectural decomposition. We intend to investigate aspects of scaleability and possibly expand the current approach through hierarchical layering and federation in the future. Note that the CM decomposition is as currently proposed by TINA [11] and does not expose the generic RCMap interfaces to its clients—again, this is for future work.

The Management Resource Map (MRMap) contains a view of all the COs instantiated in the management architecture. In fact, this instantiation takes place *through* the MRMap, by creating the relevant representation of management resources. As such the MRMap is the very first CO, necessary for the bootstrapping of the whole management system. The highest level resource manager in the MRCM domain is the Management Configurator (MC). This is "launched" by creating its resource representation in MRMap and this operation initiates the instantiation of the whole system as described later. All subsequent CO instantiations take place through the relevant resource representations in MRMap.

By its design, the MC has *a priori* knowledge of what the management architecture should consist of, and in this case it triggers the instantiation of the NTCM and CM domains through relevant RCMan COs: the NTCM Configurator (NTCMC) and CM Configurator (CMC) respectively. It also knows that the CM domain depends on the existence of the NTCM domain since the configuration of dynamic network resources depends on knowledge of the relevant static topology. As such, the MC creates the NTCMC first. The latter is responsible for the NTCM domain and creates the NTMap (Network Topology Map) and NTMan (Network Topology Manager) objects. The NTMap has initially a predefined view of network resources, reflecting the underlying network topology.

The CM domains are initialised next as the MC creates the relevant CMCs.

There may be more than one CMC and corresponding CM object group, one for each *layer network* [9, 10]. For example, in the case of ATM networks the NTMap contains static information about both the Virtual Path (VP) and Virtual Channel (VC) layer networks. As a consequence, there exist two CM domains, one addressing VP and the other VC connectivity. The MC has access to the network topology information and creates the corresponding CMCs according to the number of layer networks. Each CMC accesses topological information about its layer network and instantiates accordingly the relevant CM COs (LNC, NML-CPs, and EML-CPs).

From the moment the whole system is operational, changes to static topological information can be made by (authorized) management applications. In addition, self-configuring resources may emit notifications which are received by the NTMap and result in automatic updates of the relevant resource information. In both cases, the NTMap emits notifications which are received by the CMC for that layer network. The latter may need to reconfigure existing CM COs, launch new ones or terminate others, according to the relevant topological changes.

The generic RCMan/RCMap model can also be applied to the CM domain. In that case the relevant dynamic resource map will be distributed but a collective view of the whole range of dynamic resources will be presented through the top level RCMap object in the CM hierarchy. This federated view of dynamic resources, combined with the overall view of static resources supported by NTCM provides a consistent overall view of all network resources in the TINA system. This is a significant achievement providing a fundamental configuration management platform for many future client applications in the Management Architecture. For example, performance management activities e.g., flexible bandwidth management and (re-)distribution schemes, performance verification, capacity planning, etc. In addition, management resources allow for the controlled instantiation and termination of the management architecture. In summary, in the proposed RCM architecture there are inventory and sophisticated discovery/query/update facilities about the following resources:

- static network resources, modeling topology information;
- dynamic network resources, modeling connectivity information; and
- management resources, modeling the management components themselves.

4.2. A Prototype Implementation

We presented a general RCM architecture that addresses the resource management needs of TINA systems. A subset of the proposed architecture was specified in detail, implemented and trialed in the context of a real TINA system.

This prototype implementation served to validate and demonstrate the architectural concepts presented; as such, we describe it here.

The prototype implementation did not address the following aspects of the RCM architecture as presented in Fig. 6:

- there was no NTCMC component since the NTCM domain was very simple; relevant functionality was embodied in the MC component for simplicity;
- there was no MRMap component; instantiation of the management COs took place directly and not through a relevant resource map; and
- the CM domain was as in current TINA specifications, without a federated RCMap that offers a collective view of dynamic connectivity resources.

The components of the implemented prototype are shown in Fig. 7. The main components are the CMC, which configures the CM domain according to topological information, the NTMap which provides access to network topology information and the NTMan which allows resources to be configured dynamically. An application with a graphical user interface allows human network man-

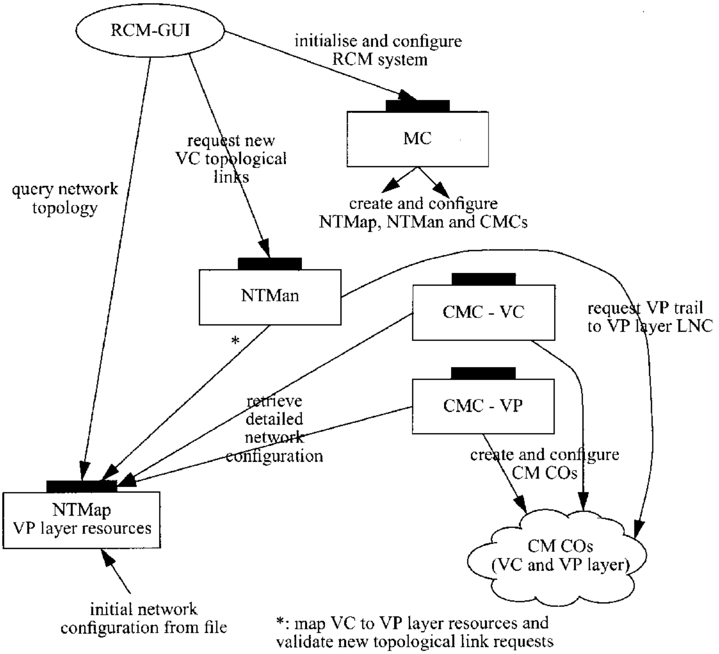

Fig. 7. RCM prototype.

agers to access and manipulate the network topology, e.g., in order to add, modify or delete topological information.

The NTMan component allows new VC layer topological links to be created dynamically. As VC layer resources are supported by VP layer resources, new topological links require the establishment of new VP layer trails. When a new topological link is requested, the NTMan queries the VC and VP layer resources in the NTMap to discover the relationship between VC layer termination point pools and VP layer termination points. After checking the consistency of the request and the availability of VP layer resources, the NTMap translates the invocation into an operation on the Layer Network Coordinator of Connection Management at the VP layer requesting a new VP trail. On successful completion, the NTMap creates the necessary set of static resources in the VC layer model in the NTMap, and informs the VC layer CMC of the new resources so that the it may configure the VC layer CM system accordingly.

The NTMap is the core of the RCM architecture since it maintains a central, consistent view of the resources in question. Given the fact that there is not yet a relevant computational interface in TINA, the NTMap computational specification is completely new. In addition, it is TMN-influenced in the sense of providing a generic CMIS-like [20] interface that enables to access information objects with various relationships. Object discovery is supported through scoping and filtering constraints in order for clients of the NTMap (such as the CMC) to build up a picture of the network resources and their relationships. As such, it constitutes a cultural difference to the TINA approach to computational specifications, as described in the next section.

The information modeling approach for the network topology was based on the TINA NRIM [10]. The fact that the NTMap computational interface is generic allows it to model different networks and instantiate the relevant system in a fashion independent of the particular network topology. The prototype system has been in fact exercized over two different ATM networks, with the NTMap initialized in a data-driven fashion. Future extensions will include interaction with the real network elements for on-line configuration purposes and the complete implementation of the presented RCM architecture.

5. USING A MANAGEMENT BROKER TO PROVIDE OPERATIONS ON MULTIPLE OBJECTS

The NTMap information model is based on the TINA NRIM, which in turn is based on the network layering concepts of [9] and the generic TMN network element information model [8]. Network resources are modeled as information objects in Quasi-GDMO (the NRIM “Network Fragment”) and should be made accessible through computational interfaces in the computational viewpoint. These information objects have various relationships and the relevant computational constructs should allow for the navigation of those relationships in a flexible fashion, e.g., to discover dynamically the network topology, provide inventory facilities, etc.

The problem is related to a methodology for mapping an information model with various relationships to a set of computational specifications. There are two distinct approaches for this mapping:

- a specific approach according to the nature of the information model in question (following possibly some general guidelines); or
- a generic approach, applicable in all cases (the *RCMap* type of computational interface).

A generic mapping results in a computational interface that provides a generic style of access while it also allows for the reuse of the relevant infrastructure. Since OSI Systems Management [14] exhibits exactly this paradigm, we have chosen to mirror its access facilities in a generic computational interface which we term a Management Broker (MB). This interface offers a CMIS-like [20] style of access in IDL and provides TMN-like access services over the TINA DPE. We have used this approach to provide the RCMap computational interface in our case study.

The specific and generic approaches discussed earlier are depicted in Fig. 8. The advantage of the generic approach is that it maps information to computational objects on a one-to-one basis and separates collective access facilities from behavioural aspects associated with the latter. These objects may be accessed through the MB, which acts as an object factory/naming server and provides multiple object access facilities based on scoping and filtering. They may be also accessed directly so that the relevant client benefits from strong typ-

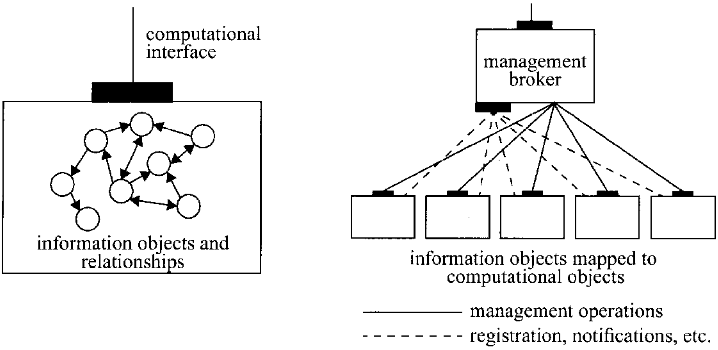

Fig. 8. Mapping information objects to computational objects.

ing with respect to the particular computational interface. The mapping of information objects in Q-GDMO onto equivalent computational interfaces in IDL is according to the guidelines of the NMF-X/Open Joint Inter-Domain Management (JIDM) group [21].

As an example, the following is a fragment of the IDL specification of our Management Broker interface. The example demonstrates two operations: the first, *objectSelection*, allows a client to identify information objects according to certain scope and filter parameters; the second, *multipleObjectGet*, allows a client to read selected attributes of a group of objects in a single operation.

```
// hierarchical naming in X.500/X.700 style

typedef struct RelativeName_t {
    AttributeId_t attrId;
    string        attrValue;
};

typedef sequence<RelativeName_t> DistinguishedName_t;
typedef DistinguishedName_t ObjectInstance_t;

// object selection through scoping and filtering

enum ScopeChoice {
    baseObjectChoice,
    firstLevelOnlyChoice,
    wholeSubtreeChoice,
    individualLevelChoice,
    baseToNthLevelChoice
};

typedef struct Scope_t {
    ScopeChoice choice;
    unsigned long level;
};
// Filter_t is a translation of the X.711 CMISFilter in IDL...

typedef struct ObjectSelection_t {
    Scope_t scope;
    Filter_t filter;
};
```

```
interface MultipleOpManagementBroker : ManagementBroker {

    void objectSelection (
        in ObjectInstance_t baseObjectInstance,
        in ObjectSelection_t objectSelection,
        out ObjectInstanceList_t objectInstanceList
    ) raises (OBJECT_SELECTION_ERRORS);

    void multipleObjectGet (
        in ObjectInstance_t baseObjectInstance,
        in ObjectSelection_t objectSelection,
       in AttributeIdList_t attributeIdList,   // optional (for all attributes)
        out GetResultList_t resultList
    ) raises (MULTIPLE_OBJECT_OP_ERRORS);

    // ...
}
```

The key advantage of the management broker approach is its genericity, which renders the MB as a server over the TINA DPE. Each MB groups together a cluster of information/computational objects while it is possible to organize those clusters hierarchically and provide a global federated view (federation issues in TMN-like object clusters have been solved through "chaining" of the relevant requests). In addition, MBs may also behave as notification servers, allowing for the fine grain control of notifications emanating from the relevant object cluster through event discriminators and filtering [22]. An additional advantage is that the MB and the relevant administered objects may be distributed as they are separate computational entities.

The benefits of providing OSI Systems Management-like facilities over the TINA DPE are manifold, as previously described. The current TINA approach is to provide them in an ad-hoc manner as required. The obvious disadvantage to this method of designing and implementing management systems is that the same features have to be re-specified and re-implemented, as required for each computational interface. There is a distinct advantage in having a generic, "standard" way of providing these. Such a generic approach needs not necessarily to rely on OSI System Management and TMN methods and techniques. The advantage, though, of doing so is that we benefit from a host of research and standardisation in this area, we are able to reuse relevant methodologies and specifications and we lay the foundation for TMN and TINA coexistence and migration strategies.

6. SUMMARY AND CONCLUSIONS

In this paper, we have proposed a generic model for Resource Configuration Management which may be applied to Network, Computing, Service and Management resources in a TINA compliant system. We have extended the scope of RCM to include Management resources, which was previously not considered by TINA. This has provided a suitable location for TINA's CMC—in the Management Resource Configuration Management area—where it was previously unclear where it should be positioned. As the management aspects of the Network Resource Architecture are expanded to cover Accounting, Performance, Fault management, etc., equivalent configurators can also be positioned within the MRCM architecture in order to provide a structured and manageable way to instantiate the relevant components.

We have shown that there are two approaches to defining configuration management interfaces to the resources being configured: generic and task-oriented interfaces, as demonstrated by our RCMap and RCMan model. The RCMan task-oriented interfaces provide high-level operations and queries on the resources which are tailored to the requirements of the managers performing the operations, and according to the specific types of resources being managed. This type of interface provides simpler operations (from the clients point of view) but loses some of the power and expressiveness of the generic approach. A severe disadvantage is that a specific component must be designed and developed for each type of manager and each collection of managed resources. The alternative approach of a generic interface, based to a large extent on OSI systems management design principles and CMIS-like operations, allows all resources to be treated in the same way with common methods for object manipulation. This approach has distinct advantages when it comes to federation: if resources are modeled in a similar way, with a common naming method, federation operations will not have to deal with a large number of different types of task-oriented interface to build a global resource map. Finally note that our approach does not preclude the use of task-oriented interfaces, it allows them to be provided by RCMan objects acting as clients of the RCMap.

We discussed our approach to providing the RCMap interface through the use of a management broker allowing operations on many objects via a single interface. As well as providing an essential service to configuration management clients, this reproduces some of the important features of OSI agents, which not only aids interaction with existing management functions in coexisting TMN systems but also eases the migration path for deploying TMN management services in TINA compliant systems.

We have specified CORBA-based management brokers that mirror the facilities of OSI systems management. These may administer clusters of other CORBA objects; they may also act as generic adaptors between CORBA clients and TMN applications in agent roles. In the former case, the TMN methodologies for producing object clusters or ensembles may be fully re-used in TINA; in fact, this is how we approached the NTMap. We have implemented a first version of such a management broker using the OSIMIS [23] platform and the Orbix implementation of CORBA. While there is plenty of ongoing research regarding TMN to TINA migration and interworking, we believe our approach retains the relevant advantages of TMN for network management while it is both compliant and complementary to the JIDM approach. In addition, this is a viable path for gradually migrating existing TMN systems over CORBA-based DPEs. We intend to propose our approach to bodies such as TINA, JIDM, OMG and the ITU-T Study Group 4.

Our view is that because the methods and techniques of the TMN have been demonstrated to be useful, and even essential to the design of complex management systems they should not be replaced without careful consideration. As we have shown in this paper, it is possible to retain many of the essential elements of the TMN, even though Q3 protocols have been replaced with IDL interfaces in the TINA DPE. The logical architecture can be kept, meaning that existing information and computational specifications can be reused. This additionally provides a smooth migration path from current (and future) TMN-based management systems to those based on TINA-like DPEs.

ACKNOWLEDGMENTS

This paper describes work undertaken in the context of the ACTS AC003 VITAL (Validation of Integrated Telecommunication Architecture for the Long term) which is partially funded by the Commission of the European Union.

We would like to acknowledge the development work undertaken by our colleagues in the VITAL project for implementing the first version of the RCM system described in this paper. In particular we would like to thank: Thurain Tin of University College London, United Kingdom, who worked on the NTMap; Lisbeth Jørgensen Veillat, Juan Carlos Yelmo Garcia and Jesus Peña Martinez of Univesidad Polytecnica de Madrid, Spain, who worked on the CMC and the graphical user interface which formed the client of the NTMap; Panagiotis Palavos of Infoline, Greece, who developed the MC; and Juan Carlos Garcia Lopez of Telefonica, Spain, who provided useful feedback on the design of the RCM system.

REFERENCES

1. TINA consortium, Overall Concepts and Principles of TINA, Document label TB-MDC.018-1.0-94, TINA-C, February 1995.

2. ITU-T X.900, Information Processing—Open Distributed Processing—Basic Reference Model of ODP—Part 1: Overview and guide to use.
3. ITU-T Q.1200 series, Intelligent Networks.
4. ITU-T M.3010, Principles for a Telecommunications Management Network.
5. ITU-T M.3020, TMN Interface Specification Methodology.
6. D. Griffin and G. Pavlou, The ICM Methodology for TMN System Design. In D. Griffin (ed.), *Integrated Communications Management of Broadband Networks*, Crete University Press, Heraklion, Greece, pp. 37–48, 1997.
7. OMG, The Common Object Request Broker Architecture and Specification (CORBA), 1991.
8. ITU-T M.3100, Generic Network Information Model.
9. ITU-T, G.803, Architectures of Transport Networks Based on the SDH.
10. TINA consortium, Network Resource Information Model Specification, Document label TB-LR.010-2.1-95, TINA-C, August 1995.
11. TINA consortium, Connection Management Architecture, Draft, Document label TB-JJB.005-1.5-94, TINA-C, March 1995.
12. J. Bloem, *et al.* The TINA-C Connection Management Architecture, *Proc. of TINA'95*, Melbourne, Australia, February 1995.
13. L. de la Fuente, *et al.* Application of the TINA-C Management Architecture. In A. Sethi, Y. Raynaud, and F. Faure-Vincent (eds.), *Integrated Network Management IV*, Chapman and Hall, pp. 424–435, 1995.
14. ITU-T X.701, Information Technology—Open Systems Interconnection—Systems Management Overview, 1991.
15. TINA consortium, Resource Configuration Architecture, Document label TB-C.AMB.001-1.0-93, TINA-C, December 1993.
16. TINA consortium, Requirements upon TINA-C architecture, Document label TB-MH.002-2.0-94, TINA-C, February 1995.
17. TINA consortium, Network Resource Architecture, Version 2.1.1, Document label NRA-v2.1.1-97-01-27, January 1997.
18. TINA consortium, Network Resource Configuration Management, Version 2.0, Archiving label EN-DK.001-2.0-96, TINA-C, October 1996.
19. TINA consortium, '94 Report on Fault Management and Resource Configuration Management, Version 1.0, Document label TR-MK.006-1.0-94, TINA-C, January 1995.
20. ITU-T X.710, Information Technology—Open Systems Interconnection—Common Management Information Service Definitions, Version 2, 1991.
21. X/Open/NMF Joint Interdomain Management specifications, GDMO/ASN.1 to CORBA IDL translation, 1994.
22. ITU-T X.734, Information Technology—Open Systems Interconnection—Systems Management: Event Reporting Management Function, 1992.
23. G. Pavlou, G. Knight, K. McCarthy, and S. Bhatti, The OSIMIS Platform: Making OSI Management Simple. In A. Sethi, Y. Raynaud, and F. Faure-Vincent (eds.), *Integrated Network Management IV*, Chapman & Hall, pp. 480–493, 1995.

Acronyms

ATM	Asynchronous Transfer Mode
CM	Connection Management
CMC	Connection Management Configurator
CMIS	Common Management Information Service
CO	Computational Object
CORBA	Common Object Request Broker Architecture

CP	Connection Performer
DPE	Distributed Processing Environment
EML	Element Management Layer
GDMO	Guidelines for the Definition of Managed Objects
IDL	Interface Definition Language
IN	Intelligent Network
ITU-T	International Telecommunications Union—Telecommunications Standardisation Sector
LNC	Layer Network Coordinator
MB	Management Broker
MC	Management Configurator
MRCM	Management Resource Configuration Management
MRMap	Management Resource Map
NMF	Network Management Forum
NML	Network Management Layer
NRCM	Network Resource Configuration Management
NRIM	Network Resource Information Model
NTCM	Network Topology Configuration Management
NTCMC	Network Topology Configuration Management Configurator
NTMan	Network Topology Manager
NTMap	Network Topology Map
ODP	Open Distributed Processing
OMG	Object Management Group
OSI	Open Systems Interconnection
OSIMIS	OSI Management Information Service
RCM	Resource Configuration Management
RCMan	Resource Configuration Manager
RCMap	Resource Configuration Map
SDH	Synchronous Digital Hierarchy
TINA(-C)	Telecommunications Information Networking Architecture (Consortium)
TMN	Telecommunications Management Network
JIDM	(NMF—X/Open) Joint Inter-domain Management task force

George Pavlou received his Diploma in Electrical and Mechanical Engineering from the National Technical University of Athens in 1982, his M.Sc. and Ph.D. in Computer Science in 1986 and 1997, respectively, from University College London. Over the last 10 years he has been doing research in the areas of protocols, performance analysis and evaluation, distributed systems, broadband network technologies, network management and service engineering. He is now a Senior Research Scientist and Lecturer and has been leading research efforts in the area of management of broadband networks and services. He has been contributing to ISO, ITU-T, NMF, OMG, and TINA and he is the author of about 30 papers in international refereed conferences and journals. In 1998 he will be joining the University of Surrey, Dept. of Engineering, as a Professor in Networking.

David Griffin received the B.Sc. degree in Electronic, Computer and Systems Engineering from Loughborough University, United Kingdom in 1988. He joined GEC Plessey Telecommunications Ltd., United Kingdom as a Systems Design Engineer, where he was the chairperson of the project technical committee of the EU RACE I NEMESYS project while working on TMN architectures and ATM traffic experiments. In 1993, Mr. Griffin joined ICS-FORTH in Crete, Greece as a Research Associate on the EU RACE II ICM project. He was the leader of the ICM group on performance management case studies and TMN systems design. Mr. Griffin joined University College London in 1996 and is currently employed as a Research Fellow working on a number of EU ACTS projects in the area of resource management for TINA systems covering performance, fault, configuration, and accounting management.