Implementing TMN-like Management Services in a TINA Compliant Architecture: A Case Study on Resource Configuration Management

David Griffin, George Pavlou, Thurain Tin
University College London, UK

Abstract

TINA aims to provide an architecture to enable telecommunications networks to support the flexible introduction of new, advanced services and to manage both the services and the network in an integrated fashion. While the specifications in the TINA Service Architecture are well advanced, network management aspects are less well defined. Resource Configuration Management is one of the most important management areas covering the management of static topology and dynamic connectivity resources; both of which are fundamental to the operation of TINA services. We present an analysis of RCM and a generic model for configuration management computational entities influenced by OSI/TMN design principles, but making use of the TINA ODP-based Distributed Processing Environment.

1. Introduction

The TINA (Telecommunications Information Networking Architecture) 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. 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) [Q1200] and the Telecommunications Management Network (TMN) [M3010] in a future integrated framework.

In this paper we are concerned with the definition, design and implementation of TMN-like functions within the TINA Management Architecture. It is a fair criticism of TINA to state that its work on the Service Architecture is much more mature than that on 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 the RCM specifications from TINA together with the requirements of a real prototype to create a design suitable for implementation.

The remainder of this paper describes the architectural issues related to the RCM management service, and demonstrates how the TINA specifications were enhanced to include Management Resource Configuration Management (responsible for the management of *management resources*), and how TMN and OSI systems management 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 main objective of the TINA consortium 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 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 modernisation of the IN. IN operation is based on control plane functions with protocol based interactions between software embedded in local switches and centralised service logic. The IN techniques have been successful for implementing enhanced telephony services, but it is more difficult to introduce modern, advanced services such as multi-media, multi-party communications mechanisms to support applications such as joint document editing. Services such as these require advanced session management and control. Strictly speaking, more complex session control *could* be provided through signalling mechanisms, protocol based interactions, and centralised 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 ODP (Open Distributed Processing) [X901] framework for specifying a ubiquitous software platform for service logic, covering both service operation and service delivery. In this way, service design and implementation can be achieved in a more flexible manner through re-usable software components. This is a revolutionary departure for the telecommunications industry and is characterised by a shift from protocol-based telecommunications engineering principles to software engineering techniques 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 [TINA-OVE]. The above discussion introduced the first two architectures: Service and Computing. The Network Architecture provides concepts for modelling the underlying network which implements the basic communications services required by the Service Architecture. TINA has based its modelling approach on the international Recommendations of the ITU, drawing on the Generic Network Information Model of M.3100 [M3100] and the SDH information model of G.803 [G803]. The resulting specification is the Network Resource Information Model [TINA-NRIM] which abstracts the communications resources forming the network infrastructure in a technology independent model.

TINA's Management Architecture draws heavily on the ITU's TMN architecture [M.3010]. The TINA specifications in the configuration management area are the most developed, especially those for connection management [TINA-CMA] [Bloem95] [DelaF95]. 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 signalling mechanisms. This is perhaps the starkest example of the paradigm shift from telecommunications to software engineering principles.[1]

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

1. However, there is an ongoing debate on whether DPE-based connection management services can perform as well as tightly engineered signalling 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. Configuration Management is concerned with managing the configuration of resources in the TINA architectures. As such, it is more specifically called Resource Configuration Management. It was thought initially that RCM should only manage the resources of the Network, Service and Computing architectures [TINA-RCA]. We believe. though, that there is scope and necessity for managing the resources of the management architecture itself, as we explain next.

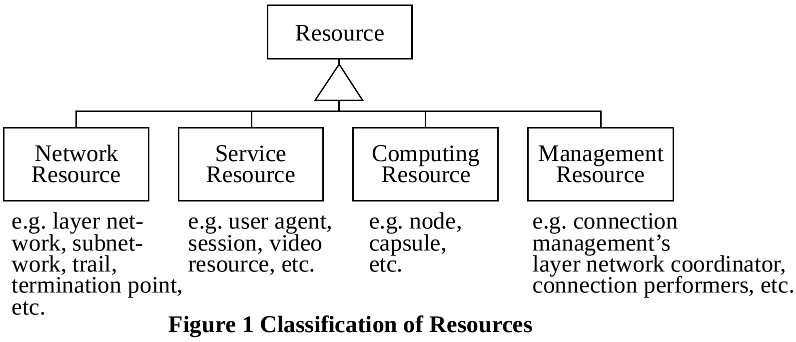

Figure 1 Classification of Resources

Layering is an important concept in TINA. Based on the TMN layering principles [M3010], TINA has defined the Service, Resource and Element layers [TINA-OVE] [TINA-REQ]. 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 specialisations 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 RCM 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 Figure 1. TINA specifications recognise the fact that within the Configuration Management 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, Network RCM (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 CM part needs to be populated with computational components which are configured according to static topological information. In fact, CM needs to be “managed” and this is done through the Connection Management Configurator (CMC) [TINA-CMA]. The latter is in fact a Resource Configuration Manager for CM 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.

The above analysis leads to the conclusion that a new domain of Configuration Management is needed, namely Management RCM (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 [TINA-NRA].

3.2 Resource Configuration Management Functions

The requirements specified in [TINA-RCA] [TINA-NRCM] [TINA-FMRCM] [TINA-NRA] and issues related to RCM in general, irrespective of whether it is Network, Service, Computing or Management RCM, can be summarised as follows:

- RCM should maintain an inventory of all resources under its influence showing relationships between the managed resources. RCM should ensure that the resource map is updated with newly installed or deleted resources.
- RCM should allow activation, deactivation, reservation and release of resources through queries and updates by other management components.
- RCM should support installation: Where physical installation is required RCM shall emit a notification.

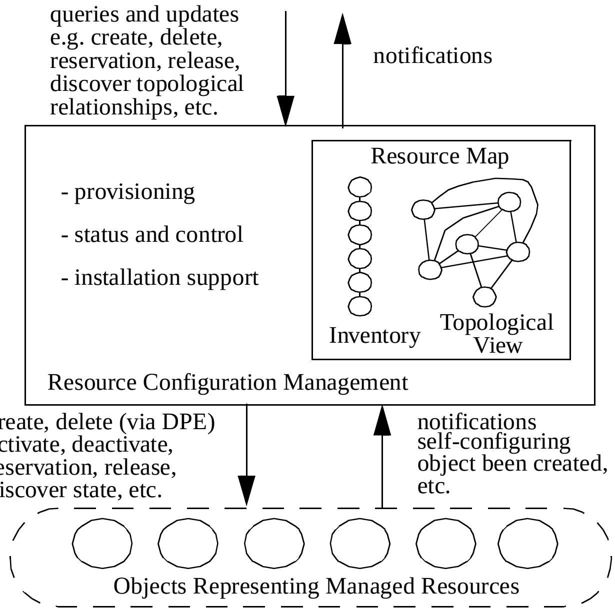

Figure 2 The Scope of RCM

Figure 2 is a representation of the scope of RCM. The figure is equally applicable to Network, Service, Computing and Management RCM.

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. Since the resource information is object-oriented, a major requirement is an object-oriented database-like access mechanism. This should provide access to information objects representing the resources and should allow their relationships to be navigated in a flexible fashion.

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 the two approaches above 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 RCM 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 behaviour 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 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.

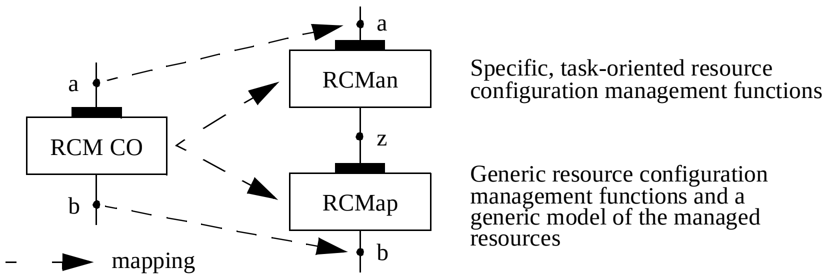

Figure 3 Specific and Generic RCM Interfaces

The RCMan/RCMap model is depicted in Figure 3. 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 these computational objects are task-oriented rather than generic as defined above. This type of computational object, generically termed RCM CO, is depicted in the left part of Figure 3. 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 modelling principles used in TINA, computational objects similar to the RCM CO may be layered hierarchically. 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 - for example, this 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
- many hierarchical RCMan/RCMap pairs, with the topmost RCMap providing a global view of the overall resource space in a hierarchical federated fashion.

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 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 [TINA-CMA] and we are reusing its architectural decomposition as currently specified.

4.1 A General RCM Architecture

The overall RCM architecture covering the NTCM, CM and MRCM domains is shown in Figure 4. The initial approach taken regarding both the MRCM and NTCM is that of a single resource map. The use of a centralised 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 fed-

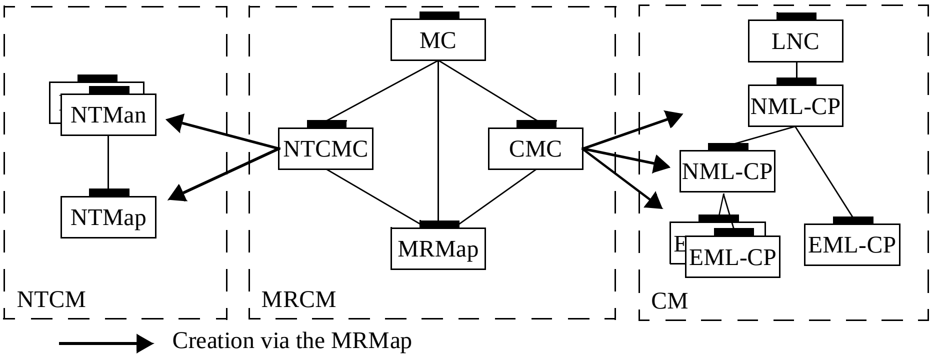

Figure 4 General RCM Architecture

eration in the future. Note that the CM decomposition is as currently proposed by TINA [TINA-CMA] 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 below. 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 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* [G803][TINA-NRIM]. 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 (authorised) 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.

4.2 A Prototype Implementation

We presented above 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 trialled 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.

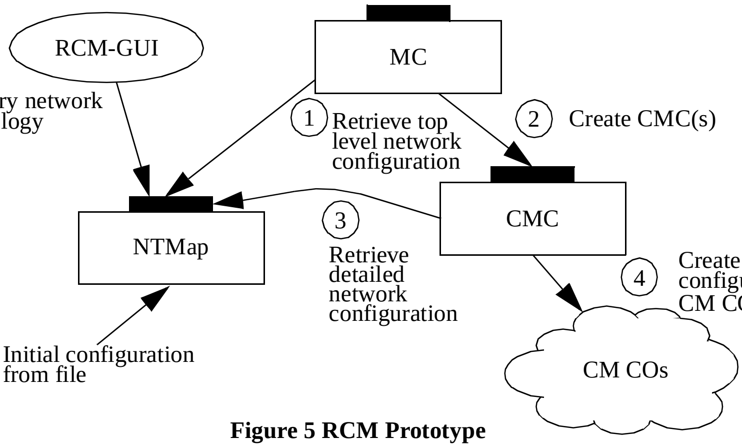

Figure 5 RCM Prototype

The prototype implementation did not address the following aspects of the RCM architecture as presented in Figure 4:

- there was no NTMan component but clients accessed directly the NTMap; in addition, there were no interactions between the NTMap and the real network elements;
- 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 Figure 5. The main components are the CMC, which configures the CM domain according to topological information, and the NTMap which provides access to the topology information. An application with a graphical user interface allows human network managers to access and manipulate the network topology e.g. in order to add, modify or delete topological information to reflect relevant network changes that took place in an off-line fashion.

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 [X710] 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 modelling approach for the network topology was based on the TINA NRIM [TINA-NRIM]. 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 exercised over two different ATM networks, with the NTMap initialised 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 RCM architecture.

5. Using a Management Broker to Provide Operations on Multiple Objects

Network resources are modelled 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.

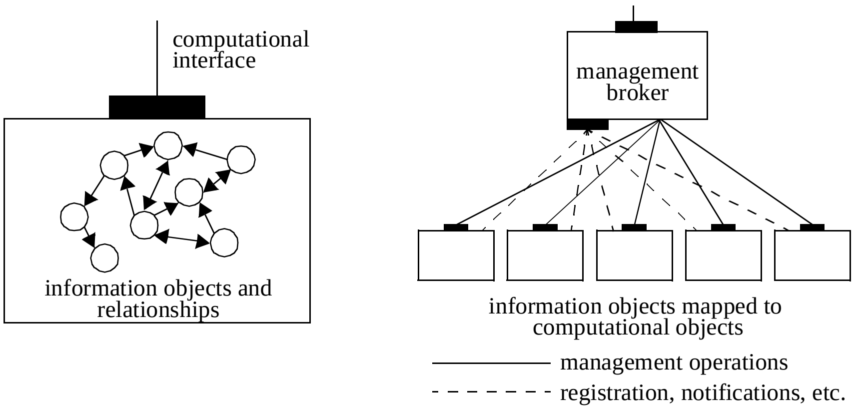

Figure 6 Mapping Information Objects to Computational Objects

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 [X701] 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 [X710] style of access in IDL and provides TMN-like access services over the TINA DPE. We have used the MB approach to provide the RCMap computational interface in our case study.

The specific and generic approaches discussed above are depicted in Figure 6. 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 typing 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 [JIDM].

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 organise 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 [X734]. 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 many, as described above. The current TINA approach to these facilities 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.

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. The alternative approach of a generic interface, based to a large extent on OSI systems management and CMIS-like operations, allows all resources to be treated in the same way with common methods for object manipulation.

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 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 fully 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 standards 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 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.

7. Acknowledgements

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: Lisbeth Jørgensen Veillat, Juan Carlos Yelmo Garcia and Jesus Peña Martinez of UPM who worked on the CMC and the graphical user interface which formed the client of the NTM; Panagiotis Palavos of Infoline who developed the MC; and Juan Carlos Garcia Lopez of Telefonica who provided useful feedback on the design of the RCM system.

This paper describes work undertaken in the context of the ACTS AC003 VITAL project. The ACTS programme is partially funded by the Commission of the European Union.

8. References

[Bloem95] Bloem, J., et al, The TINA-C Connection Management Architecture, TINA'95, Melbourne, Australia, Feb. 1995.

[DelaF95] de la Fuente, L., et al, Application of the TINA-C Management Architecture, in Integrated Network Management IV, pp. 480-493, Chapman & Hall, 1995.

[G803] ITU-T, G.803, Architectures of Transport Networks Based on the SDH.

[Gri95] Griffin, D., Georgatsos, P., A TMN System for VPC and Routing Management in ATM Networks, in Integrated Network Management IV, pp. 356-369, Chapman & Hall, 1995.

[Gri97] Integrated Communications Management of Broadband Networks, pp 37-48, Chapter 3, The ICM Methodology for TMN System Design, Griffin, D., editor, Crete University Press, ISBN 960 524 006 8, 1997.

[JIDM] X/Open/NMF Joint Interdomain Management specifications, GDMO/ASN.1 to CORBA IDL translation, 1994.

[M3010] ITU-T M.3010, Principles for a Telecommunications Management Network.

[M3020] ITU-T M.3020, TMN Interface Specification Methodology.

[M3100] ITU-T M.3100, Generic Network Information Model.

[Pav95] Pavlou, G., Knight, G., McCarthy, K., Bhatti, S., The OSIMIS Platform: Making OSI Management Simple, in Integrated Network Management IV, pp. 480-493, Chapman & Hall, 1995.

[Q1200] ITU-T Q.1200 series, Intelligent Networks.

[TINA-CMA] Connection Management Architecture, Draft, Document label TB_JJB.005_1.5_94, TINA-C, March 1995.

[TINA-FMRCM] ‘94 Report on Fault Management and Resource Configuration Management, Version 1.0, Document label TR_MK.006_1.0_94, TINA-C, January 1995.

[TINA-MA] Management Architecture, Version 2.0, Document label TB_GN.010_2.0_94, TINA-C, December 1994.

[TINA-NRA] Network Resource Architecture, Version 2.1.1, Document label NRA_v2.1.1_97_01_27, January 1997.

[TINA-NRCM] Network Resource Configuration Management, Version 2.0, Archiving label EN-DK.001_2.0_96, TINA-C, October 1996.

[TINA-NRIM] Network Resource Information Model Specification, Document label TB_LR.010_2.1_95, TINA-C, August 1995.

[TINA-OVE] Overall Concepts and Principles of TINA, Document label TB_MDC.018_1.0_94, TINA-C, February 1995.

[TINA-RCA] Resource Configuration Architecture, Document label TB_C.AMB.001_1.0_93, TINA-C, December 1993.

[TINA-REQ] Requirements upon TINA-C architecture, Document label TB_MH.002_2.0_94, TINA-C, February 1995.

[X701] ITU-T X.701, Information Technology - Open Systems Interconnection - Systems Management Overview, 1991.

[X710] ITU-T X.710, Information Technology - Open Systems Interconnection - Common Management Information Service Definitions, version 2, 1991.

[X734] ITU-T X.734, Information Technology - Open Systems Interconnection - Systems Management: Event Reporting Management Function, 1992.

[X901] ITU-T X.900, Information Processing - Open Distributed Processing - Basic Reference Model of ODP - Part 1: Overview and guide to use.