22

Experiences in Integrated Multi-Domain Service Management

D. Lewis, T. Tiropanis, A. McEwan
Department of Computer Science, University College London
Gower Street, London, WC1E 6BT, United Kingdom
tel: +44 171 391 1327, fax: +44 171 387 1397
E-mail: D.Lewis@cs.ucl.ac.uk, T.Tiropanis@cs.ucl.ac.uk, A.McEwan@cs.ucl.ac.uk

C. Redmond, V. Wade
Department of Computer Science, Trinity College Dublin
O'Reilly Institute, Dublin 2, Ireland
tel: +353 1 608 1765, fax: +353 1 677 2204
E-mail: Cliff.Redmond@cs.tcd.ie, Vincent.Wade@cs.tcd.ie

R. Bracht
IBM, European Networking Centre,
Vangerowstr. 18, D-69115, Heidelberg, Germany
tel: +49 6221 59 4462, fax: +49 6221 59 3300
E-mail:R.Bracht@heidelbg.ibm.com

Abstract

Increased competition, complex service provision chains and integrated service offerings require effective techniques for the rapid integration of telecommunications services and management systems over multiple organisational domains. This paper presents some of the results of practical development work in this area, detailing the technologies and standards used, the architectural approach taken and the application of this approach to specific services. This work covers the integration of multimedia services, broadband networks, service management and network management, though the detailed examples given focus specifically on the integration of services and service management.

Keywords

Multi-domain service management, TINA Service Architecture, reusable service components

Management of Multimedia Networks and Services R. Boutaba & A. Hafid (Eds.)

1. INTRODUCTION

In the increasingly competitive telecommunications market, service providers must develop and deploy user services and their management systems at increasingly faster speeds. At the same time these systems become more complex as different user services, including those from different providers, are combined into integrated service offerings. This requires the integration of service software components as well as the integration of the management components that manage user services. Using industry agreed interfaces for such integration allows third party service software developers to provide "off the shelf" reusable components for management and service software. This approach however, may prove problematic where competitive pressures force the pace of change beyond that at which industrial agreements can be readily made. In this case common management and service components intended for reuse in different service systems will require appropriate flexibility in order to suit the requirements for new services as they emerge.

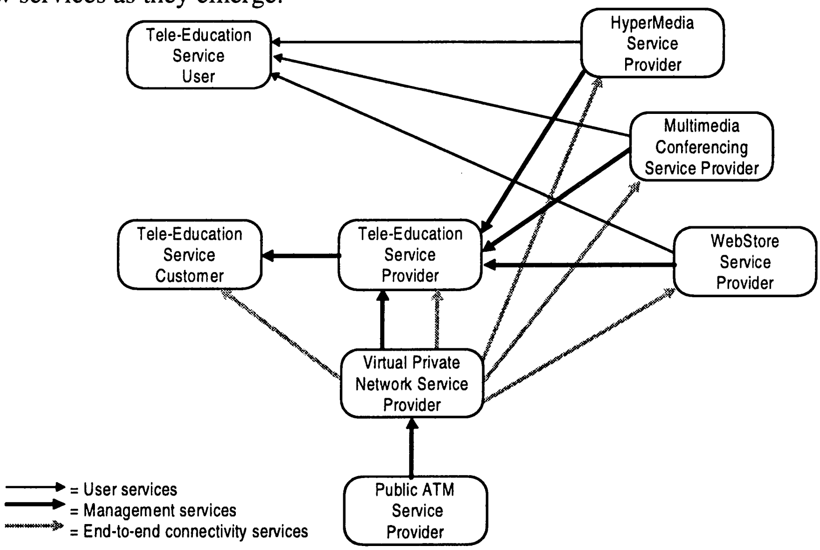

Figure 1 Prospect Stakeholder Model

Many of these issues have been addressed in the EU sponsored project Prospect which is examining the integration of telecommunications services and management in the open service market. The particular business situation adopted in Prospect (see figure 1) provides a business scenario intended to be complex enough to generate realistic service integration problems. This scenario is based on a provider of educational courses that offers a Tele-Education Service (TES) to its customers. This service is an integration of three examples of a Multimedia Tele-Service (MMTS, i.e. a distributed, multimedia telecommunications service); a Hyper-Media (HM) service, a Multimedia Conferencing (MMC) service and a

Web-Store (WS) service. Each MMTS is operated by a separate provider organisation. In addition the TES provider subscribes to a general purpose Virtual Private Network (VPN) service that it uses to provide broadband network connectivity between itself, its customers and the MMTS providers. The VPN provider further extends the service chain by using the ATM Virtual Path management service offered by a public network provider, which delivers international links between the connected customer and provider sites.
The following section presents some of the technologies and architectures adopted for implementing systems in this business model. Section 3 describes the overall approach to service and management integration in Prospect, and provides details on some specific design areas before concluding.

2. ARCHITECTURES AND TECHNOLOGIES

In investigating the integration of telecommunication services and management a pragmatic approach was taken, both to facilitate the development of a working trial and to ensure that the guidelines developed would be relevant to both existing and evolving technologies, architectures and services. The selection of the approaches adopted is summarised below.

2.1. Architectures: TMN and TINA

Though OSI-SM principles and standards have been applied successfully at the network and network element layers of the TMN logical layer architecture [Hall], they have not penetrated far into the service management layer. This is due to several factors. First there is a general lack of standards related specifically to service management, through more recently this is being addresses by the Network Management Forum's (NMF) SMART working group [Adams]. There is also a lack of a common multimedia service model on which to base service management functional definitions, i.e. there was no single industry agreed model of what was to be managed. Finally, the diversity of functionality and types of system that are the target of service management impose more complex requirements on software engineering solutions than are typically found in network management problems. So while a TMN-standards based approach for the ATM VP service provider was retained in Prospect, other approaches were sought for structuring and engineering the integration of multimedia services and service management.
The Telecommunications Information Networking Architecture (TINA) Consortium has been specifically addressing the area of telecommunications services and their management. For this, TINA has developed a service architecture [TINA-SA], a TMN-based management architecture [TINA-MA] and a network architecture (influenced by ATM Forum and TMN network models). These architectures provide the consistent, detailed model for the

integration of service, management and network components that are out of the scope of existing TMN recommendations.
Therefore, the integrated service and management model of the TINA Service Architecture (SA) was selected as the basis of the TES and MMTS providers' systems as described in the section 3. TINA's integration of the network architecture with the service and management architectures was not, however, adopted, since it assumes a call-based model that cannot be straightforwardly resolved with the connectionless communication model used by the Internet applications used in Prospect (see section 2.3). The information technology approach taken in TINA is based on a distributed processing environment (DPE) for which CORBA was adopted here.

2.2. Software Integration Technologies: CORBA and Java

Recent development in distributed systems technologies have been led by the Object Management Group [OMA], and in particular their definition of the CORBA 2.0 standard. CORBA 2.0 allows for the specification of object interfaces and their abstract location in a (virtual) distributed environment. Object interface definition is separated from object implementation, thus allowing components of a CORBA 2.0 compliant object network to use different (possibly proprietary) implementation languages. The freedom this imparts to the developers of components, intended for integration into large systems, ideally places this technology for use in the open services market. In Prospect CORBA was used primarily for implementing computational objects derived from the TINA SA. These objects typically had multiple interfaces, which were mapped onto individual CORBA objects.
The user interface systems developed have some specific requirements centred around the close integration of components and portability, i.e. the user interfaces have to integrate the interfaces offered by the different services being made available to users in an apparently seamless way on both UNIX and Win32 platforms. The definition of an IDL mapping for Java in the CORBA 2 standard allows for this language to be used to implement desktop service client components for users. The portability of Java and the well defined inter-applet communication interface meant that this was the most viable solution. In addition, Java provided a path to the integration of the user interface components with WWW browsers, either as Java applets or as Java implementations of Netscape plug-ins, and to the dynamic distribution of service client component software. The Java AWT library also was used to provide integrated, portable graphical user interfaces.

2.3. Network Technologies: Internet and ATM

The focus of Prospect is on an open market for multimedia broadband services. ATM is often assumed to be the primary technology to provide such services, yet the lack of commercial, switched wide area ATM services and the absence of any killer applications is delaying its widespread adoption. In comparison, Internet technology is the major solution for open digital communications today, with a huge growth in commercial Internet service providers and a wide range of applications and services being available. Two of the major developing strands of Internet services provide the basis for the MMTS:

The M-BONE, though still largely a non-commercial trial service, has shown the feasibility of real time, multiway, multimedia communications. This technology is easily scaleable from small to large groups of communicating individuals and makes optimum use of network resources. This provided the basis for the MMC service.

The World Wide Web, due to: the ease of providing information, the free availability of browsers and the use of a simple communications protocols has enabled providers to develop and deploy new services in an open way, thus allowing a wide range of commercial services to be offered over telecommunications networks. This therefore provides the basis, in different variants, of the other MMTSs.

Prospect has elected to use existing IP applications between hosts on mixed technology LANs, communicating via routers interconnected over a backbone of campus and wide area ATM networks. Such a configuration provides an upgrade path to the integration IP and ATM technologies, as they converge in their support for QoS guarantees, e.g. as in IP switching. It is also typical of current ATM usage and therefore provides a realistic and practical basis for broadband user trials.

The required IP connectivity was provided in the form of a multi-domain Intranet (sometimes termed an "Extranet"), in this case connecting customer, TES provider and MMTS provider LANs. This was controlled in a TMN-structured management plane, separate to the control of the services themselves, i.e. the service sessions did not interact directly with communication sessions in the way prescribed in TINA. This was primarily due to the lack of manageable signalling technology that could be incorporated in the trial. Instead all ATM network level configuration was managed by TMN systems to configure semi-permanent ATM links which operated as the back-bone of the IP network. Both this backbone and the configuration of IP routers, LANs and campus networks, were handled by the VPN service. This aimed to hide from its customers (i.e. the TES and MMTS providers) the complexity of this technologically heterogeneous, multi-domain network management problems, by offering management functionality via a network technology independent CORBA interface. This required the VPN provider to also handle the mapping from CORBA technology domain to the different technology domains used for managing networks in the various different administrative domain over which it provided a service. For the ATM service provider domain, which offered a CMIP interface to its network service, a static CORBA-to-CMIP gateway was developed based on NMF-X/Open's Joint Inter-Domain Management task-force recommendation for IDL-GDMO translation [X/Open].

3. INTEGRATION MODEL

3.1. Common Service and Management Components

The enterprise model contains several organisations providing different types of services, however many of the user and management activities that need to be performed by the systems used by customers and providers are similar, and therefore lend themselves to common solutions. This was appropriate for the TES and MMTS providers who all have to solve similar problems with the control and accounting of end user access to their services. The approach taken to these problems involved the development of common component interfaces and implementations that were reused in the different systems of the various service providers.

The TINA SA integrates the end user's access to and interaction with a telecommunications service with the management of service-related subscription and accounting information. This architecture identifies areas that are regarded as service specific, i.e. those that will be different for different service offerings, and those which are service independent, i.e. those that will remain constant over different service implementations. In this way, the SA model provides a core set of common inter-operable interfaces between its components, as well as a common semantic model around which services can be developed, both of which aid in the integration of separate services. The TINA SA was therefore deemed well suited as a basis for common service and management components. The version of the SA publicly available when this work was conducted, i.e. version 2.0, assumed, however, that all services were supplied by a single provider. This architecture therefore had to be augmented in order meet the multi-domain requirements of the Prospect business model.

The major common components of the Prospect design model derived from the TINA SA are; the integration of service control in the user and provider domains, termed the Desktop Service Integration (DSI) component; the Subscription Management component and the Accounting Management component.

The DSI component is based on the TINA separation of issues related to user access to services and issues related to usage and control of the service. These are modelled in TINA as separate access and service sessions, but augmented here to operate with current desktop technologies and to use Internet protocols for service data flow, rather than an explicitly controlled communications session as prescribed in TINA. The aim was to provide a single model that each of the providers could use, but which could be easily integrated at the desktop in a way that would render the multi-provider composition of the services transparent to the end user. This required a clear distinction between the parts of the model which were service independent and could therefore be used directly by each service provider, and parts of the model that were service specific, and were thus expected to be augmented by the individual service providers to accommodate functionality particular to their service.

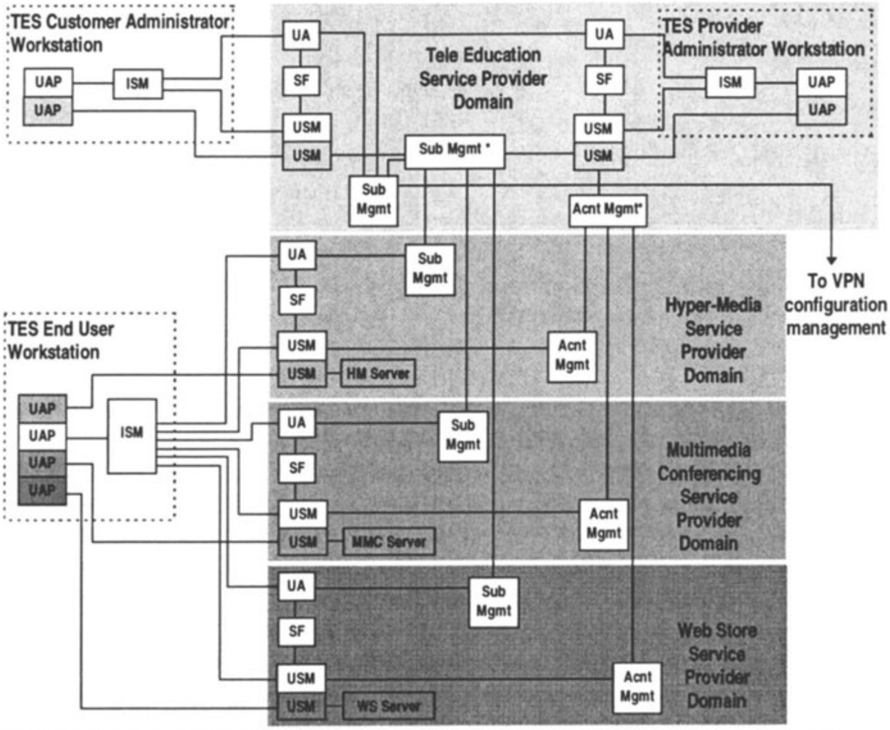

Figure 2 Integration of Service independent and service specific components

In the user domain the DSI model modifies the TINA SA to support the integration of existing applications, adopting a component-based approach using Java applets integrated into a WWW browser which together form the user application (UAP) component of the model. Also in the user domain a separate object, named the Integrated Session Manager (ISM) mediated all the service independent interactions between the user and provider domain. This provided additional support for hierarchical session relationships, thus supporting the composition of service sessions from different providers, into what appears to the user to be a single service session.

In the provider domain the TINA SA concepts of a User Agent (UA), Service Factory (SF) and User Service Session Manager (USM) have been adopted, providing the service independent components that allow the user domain components to interact correctly with the Subscription and Accounting Management components. The TINA service session concept has however been simplified by removing the requirements for group session functionality. Where needed, this functionality is provided by service specific components, e.g. multicast groups in the MMC service. The UA mediates between the user and the Subscription Management component of the provider's system, relaying the services the user may access and allowing the user to initiate a session in one of those services. The SF is then used by the UA to create a USM that allows the user to interact with a service session. The TINA SA service session interactions, e.g. initiate, terminate, suspend and resume, are handled by the service independent parts of the USM, including logging them with the usage metering parts of the Accounting Management component. Other service specific interactions need to be implemented by the service provider responsible, as extensions to the USM and corresponding UAP components.

The Subscription Management component allows for the authorisation and barring of users to specific services, and the addition and removal of network sites supporting these users. Parts of the Subscription Management component are identified as areas for service specific specialisation; for instance the subscription registrar (SubRgs) component, which performs service specific operations on the basis of service independent ones it receives. The Accounting Management components allows service usage monitoring through interactions with a USM, charging based on tariffs agreed for a subscription, and bill generation based on these charges and a customer specific tariff plan which specifies discounts, payment schedules etc..

Figure 2 shows how the various service independent components (white boxes) and service specific components (shaded boxes) have been combined in integrating the service and management systems of the TES and MMTS providers. Within the project, these common components were implemented by one set of partners, and then reused by other partners in the implementation of the specific service provider systems they will operate. This simulated, to an extent, the development environment of the open service market which provided a means of gaining direct experience of some of the problems involved. All the objects shown in figure 2 were implemented on a single vendor's CORBA 2.0 compliant platform, i.e. Orbix, OrbixWeb and OrbixNames, to avoid any potential IIOP interoperability problems. Objects were actually implemented in either C++ or Java. The following sections describe in more detail how common components were used for specific services.

3.2. Integration of the Tele-education Service

The TES system is a good example of how a service can be constructed from an integration of existing services and common, reusable service and management software components. The integration for this service offering can be decomposed into the integration of the user services and the integration of management services.

The integration of user services was performed in the user application. This was based on a WWW browser that integrated the Java applets which performed the service independent UAP functions of the DSI components as well as the applets supplied by the MMTS providers to perform their service specific interactions. The problem of integrating the services was therefore reduced to that of defining inter-applet communications. This enabled the TES UAP developer to concentrate more on the look and feel of the user interface, and the composition of the MMTS in delivering the TES to the user, rather than how the UAP interacted with the different MMTS providers' systems.

The development of the TES management system involved both the development of provider domain management systems and mechanisms for the TES administrator and TES customers to interact with these systems. The TES needs to co-ordinate its management with that of the subcontracted MMTS and VPN providers. For subscription management this required propagating operations such as the activation and deactivation of service profiles and the authorisation and barring of end-users, to the Subscription Management components of the MMTS provider. This was done by developing a wrapper (SubMgr* in figures 2) for the TES provider's common Subscription Management component, that forwarded the relevant operations to the identical common components in the MMTS providers' domains. In addition the SubRgs CO of the TES was modified to perform the correct configuration management operations when a customer site was added or removed, i.e. to use the VPN provider's configuration management services to connect to this site. For accounting management the interactions with sub-providers was limited to the collection of billing information from the common Accounting Management components used by each MMTS provider.

The customer and provider management UAPs were based the common DSI component. For the provider the UAP and the USM were extended to provide the user with access to subscription and accounting management information as well as management functions controlling how this information is propagated to specific sub-provider systems. For the customer the UAP and USM provided a subset of these operations, allowing the customer access only to specific operations related to their particular service subscriptions.

3.3. Integration of the Hyper-Media Service

The Hyper-Media service is a WWW based service, modified to only allow session based access. The TINA-based session mechanism allows easier service access control and service accounting, and fits well with the TINA model.

Customisation facilities can also be offered to customers because record of their information traversal path is kept in the service provider domain. The generation, storage and access of per-user data in the service provider domain is supported and can be used to provide functionality that is currently dependent on browser implementation. Common examples of such per-user data include bookmarks and history lists.

Although the concept of a session is used, its realisation within an inherently connectionless service was only made possible by the de facto support of HTTP state mechanisms, 'Cookies', in many common client implementations. The general usefulness of a state mechanism in HTTP has been recognised and there are now Internet standards track documents relating to this area [RFC2109], as well as other work in progress. This application protocol session support is independent of the persistent connection ('Keep alive') concept supported in most HTTP client and server implementations, where the same TCP connection is kept open for multiple HTTP requests.

Realising a session using additions to the HTTP protocol also caused other problems; many users will, and do, access information services through firewall proxies that normally cache the information being retrieved to allow other users making the same request access the cached copy. This mechanism can help alleviate congestion on the Internet, but to an information seller, which may have many customers in the same organisation (all using the same cached copy) this can reduce revenues on per copy charges. Most caches also cache the 'Cookie' values returned to the original requester, reducing the ability to track individual sessions. To get around this difficulty a proxy is used to ensure that each user accesses a document using a unique URL, ensuring that any intervening caching proxy only caches a document for that user's own use. The proxy mechanism also gave other benefits, namely an access and session control capability.

The main service server, a modified Apache daemon [APACHE] is either kept behind a firewall, or configured to only accept requests from its own domain. A user can only access this server through the TINA desktop components that will initialise a session for that user. In the course of the session initialisation the USM creates a proxy on a firewall machine that directs requests for that user to the main server inside the firewall. The host, port and the path to a 'Cookie' generating script are combined to form a URL that is returned via the UAP applet to the browser. The user has then established a session with the service. Denying service access to the user is a simple matter of destroying the proxy in the service domain. Ending a session will destroy the proxy in the normal way. The proxy

mechanism could also enable service announcements by directing all requests to another daemon inside the firewall.
Accounting is accomplished by a CORBA one way call embedded in the user tracking module of the daemon to a service dependent interface of the USM. From the USM the usage data is mapped to requests inside the service session and is forwarded to an implementation of the TINA UMData. The usage data collected can be used in the usual way for charge and bill generation, following logging at the end of a session.

4. CONCLUSION

This work provides an example of how the definition of common, reusable components aids in the rapid development of integrated telecommunications services. It was found important to define clearly which parts of a reusable component are intended to be used as-is, i.e., the service independent parts, from those that must be modified by its users to integrate into a particular service, i.e. the service specific parts. The widespread use of common components was also found to ease the task of developing integrated services, since the common parts can be designed explicitly to support such integration.
Though this project does not aim to be TINA compliant, the TINA service architecture was found to provide a practical framework for controlling and managing multimedia service offerings based on IP technology. However the connection-based communication session and group session semantics from TINA were not found applicable to the IP domain.
CORBA was found to provide a suitable basis for implementing TINA-based components, though fuller use of CORBA Common Object Services, e.g. the life-cycle service, is required. From this work, it also appears that a combination of WWW, Java and CORBA provides a practical approach to presenting integrated services to the user. However a suitable security model for applets that need to communicate with multiple remote objects needs to be supported in WWW browsers.

REFERENCES

[Adams] Adams, E., Willetts, K., The Lean Service Provider: Surviving the Shakeout Through Service Management Excellence, McGrw-Hill, ISBN 0-07-070306-X, 1996

[APACHE] http://www.apache.org/

[Hall] Hall, J. (editor) Management of Telecommunications Systems and Services: Modelling and Implementing TMN-Based Multi-Domain Management, Lecture Notes in Computer Science 1116, Springer-Verlag, ISBN 3-540-61578-4, 1996

[OMA] Object Management Architecture Guide, ed. R.M. Soley, OMG Document Number 92.11.1, Revision 2.0 Second Edition, Object Management Group, 1992

[RFC2109] ftp://ds.internic.net/rfc/rfc2109.txt

[TINA-MA] de la Fuente, L., A., Walles, T., Management Architecture, TINA Baseline Document TB_GN.010_2.0_94, December 1994

[TINA-SA] Bernt, H., Kim, C., Kim, L., et al., Service Architecture, TINA Baseline Doc. No. TB_MDC.012_2.0-94, TINA-C, March 1994

[X/Open] Inter-domain Management Specifications: Specification Translation, X/Open Preliminary Specification, Second Sanity Check Draft, June 1996

BIOGRAPHY

Dave Lewis graduated in Electronic Engineering at the University of Southampton in 1987 and received an MSc in Computer Science from UCL in 1990, since when he has worked as a research fellow. He is also working on a Phd, researching a service management development architecture for the open services market.

Thanassis Tiropanis is a research fellow at University College London. He graduated from the department of Computer Engineering and Informatics at the University of Patras, Greece in 1993. The management of mobile services in a distributed environment is his particular area of interest in which he is doing a Ph.D.

Alistair McEwan is a researcher into Distributed Computing, his interest coming from both Software Engineering and HPC perspectives. Previous work in the area of distributed computing includes the development of heterogeneous message passing libraries and Software Engineering systems for implementing Parallel and Distributed Systems in C++.

Cliff Redmond received his Bachelors degree in Computer Engineering from the University of Dublin, Trinity College, in 1995. His MSc research is concentrated on information service accounting issues. Other research interests include component based software engineering and systems management.

Vincent P. Wade is a lecturer in the Computer Science Department, Trinity College Dublin. He originally graduated from University College Dublin with a Bachelor of Computer Science in 1987 and received his MSc in Computer Science from Trinity College Dublin in 1991. His research interests include network and service management systems, advanced multimedia information services, platforms for distributed systems.

Ralf Bracht received a Ph.D. in physics from Ruprecht-Karls University, Heidelberg, Germany, in 1995. Since then he has been working at IBM Science and Technology Center in Heidelberg. He is project manager at IBM and Prospect's technical chairman. His current interest is management of telecommunications services.