UNIFIED MESSAGING

A Framework Approach for building messaging services

Timo HäMäläinen and Esa Saukkonen
Centia Ltd., sponsored by Hewlett-Packard

Key words: Unified messaging, Value-added services, Frameworks

Abstract: This paper presents an approach for the development of a unified messaging (UM) solution, which enables users to combine their messages into a single virtual mailbox, enabling message access, response and administration from anywhere, at anytime.

1. INTRODUCTION

Competition in the mobile market has now reached the stage where falling margins from traditional basic services are impacting upon the revenues of most European operators. As price differentials between operators in national mobile markets are reduced, the introduction of carefully priced, sophisticated value added services is seen as a prime alternative revenue source and a key factor in winning and retaining customer loyalty. One service considered by most operators to be of critical importance for customers is that of message management.

Increasing competitive pressures and rising costs are forcing enterprises worldwide to improve the speed and efficiency with which they do business with their customers. Often this means equipping a mobilised work force to enable rapid communication with customers and clients from anywhere at any time but, for many organisations, this task is complicated by the variety of message services employed. The mixture of fax, voicemail and email services requires that multiple devices are required to send, respond to and administer these messages, resulting in increased maintenance costs and, frequently, in lost revenues due to late response times.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35581-8_35

T. Yongchareon et al. (eds.), *Intelligence in Networks*

While the scope for outsourcing among Small to Medium sized Enterprises (SME's) is not so broad as for large corporations, re-organisation of the office administration and communication functions provides equally significant opportunities to enhance profitability through improved efficiency. Most SME's, however, do not have the capability to have their own messaging services because of the relatively high cost of these solutions. Because of this reason, service provider offered solution best fits for the SME sector.

This paper presents an approach for the development of a unified messaging (UM) solution which enables users to combine their messages into a single virtual mailbox, where they can be accessed and processed anywhere, at anytime.

2. REQUIREMENTS FOR UM SOLUTION

As mentioned earlier, the term unified messaging describes the nature of the resulting services, where different message types are combined into a single virtual mailbox, which can be accessed with multiple types of methods. Virtual mailbox can be achieved by collecting messages from various sources into a central location. Multiple access methods can then be obtained by connecting this 'central mailbox' into different kinds of access networks like Internet or telephony network.

It is also important to note that when there is a way to access many types of messages, messaging can be made more intelligent. This means that UM solution should provide for example ways to convert messages from one type to another and ways to notify people who don't have temporarily email client available.

2.1 A closer look at interfacing need

UM solution must interface with the various PSTN, PLMN and Internet components, which are needed to receive and deliver messages from and to other messaging systems. Some of these interfaces conform to international standards but others are proprietary systems. The following diagram details some of the more important components with which the UM solution must interface.

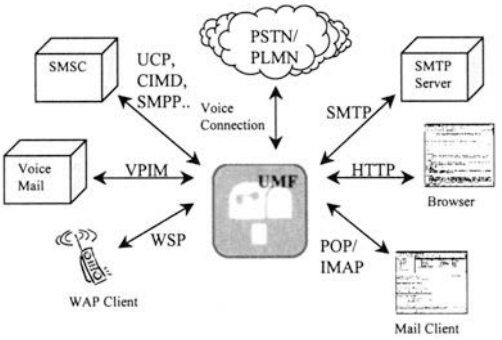

Figure 1. Interfacing with other systems

The most common messaging systems today are email and voicemail. In order to receive email messages the UM solution must support SMTP protocol, which is used between Internet mail servers to deliver mail messages, and therefore act like a normal SMTP server. The UM system must also interface with other email servers which employ POP or IMAP protocol – and thus behave like a normal email client, enabling message retrieval from other possible mailboxes of a user to a central place.

Retrieval of voicemail is much more complex issue because voicemail systems do not normally provide external interfaces. This issue will be hopefully resolved with the introduction of VPIM (Voice Profile over Internet Mail) standards, which define how to transport voice and fax messages over SMTP protocol. An alternative solution is to build voice mail capabilities into the UM system, which means that UM system must interface with telephony network.

Message notification is a critical feature for any messaging system. In the GSM environment the short messaging feature provides a convenient method to deliver notification and it can also be used to send the actual messages and to carry mailbox management information. Using of short messages can be motivated with fact that they have been used to implement many kinds of information retrieval and other value-added services in Scandinavia. Short message systems interfaces are often proprietary, the most common of which are UCP, CIMD, GPI and SMPP.

The most common way of accessing messages is an email client. Web browsers can also be used to access messages and, in the near future, access methods such as Wireless Application Protocol (WAP) can be used to run a micro web browser on mobile phones providing the necessary mailbox access. Besides of accessing messages, web browsers can be used to manage UM system configuration. Configuration management with web browser is useful especially for users to modify their private configuration information,

like passwords since browser is commonly accepted and commonly available information access tool.

3. FRAMEWORK APPROACH AS A UM SOLUTION

Today most of the value added messaging services are being built one by one resulting fast time to market for the first few services. Here, every messaging service setup is custom project, thus resulting to unacceptable service creation speed in longer term. Also, individual services require vast management resources. The framework approach offers fast time to market for new messaging services through utilisation of existing resources.

The UM solution described in this paper comprises UM services and UM framework (UMF). UM framework is a collection of components that simplify application development and run-time organisation. UM service applications together with UM framework provide the functionality which is visible to the users of UM services.

4. UM FRAMEWORK LOGICAL ARCHITECTURE

This section describes the logical architecture of the UM framework (UMF) which not only presents a unified method of accessing different kinds of messages but also offers a sophisticated platform for service providers to develop a wide range of UM services.

The following schematic outlines the architecture and the environment of the UM framework. This logical architecture is not, however, limited to unified messaging and can be applied to many kinds of network service platforms since it is not bound to specific networks or services.

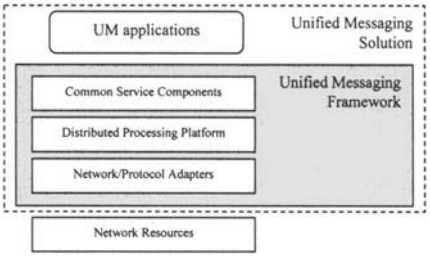

Figure 2. Architecture of the UM framework

The UMF environment comprises UM applications and network resources. UM applications access and manipulate different kinds of

messages and they can be built by harnessing the components provided by the framework. It is the UM applications which implement the unified messaging functionality that can be seen by the users of the UM services.

A typical UM application would be a 'smart mailbox with SMS notification' which is simply a mailbox, which sends a notification about an incoming email to the receiver by using a GSM short message. This application could also be employed to automatically retrieve emails from the user's other mailboxes to a central location because people usually have multiple mailboxes. When a user receives SMS notification, one can retrieve the mail from POP email server application by using POP compatible email client or use text-to-speech email application to listen the mail.

Network resources represent different kinds of networks, which can be used to access the UM applications. These resources can also be employed by the applications since their functionality can contain network communication like in the case of SMS notification described earlier. Because the unified messaging in general gives a unified view into a message box containing different types of messages from various kinds of clients, different kinds of networks and protocols must be adapted to the framework.

4.1 UM framework layers

This section describes various layers of the UM framework. The components of UMF have been categorised into layers according to their functionality:

- Distributed processing platform enables communication between the components of UMF.
- Network/Protocol adapters connect the framework to different networks.
- Common service components are reusable functional blocks which are being utilised by the UM applications.

4.1.1 Distributed processing platform

The UMF and UM applications consist of components which interact with each other. The distributed processing platform (DPP) provides support for their execution and interaction in the distributed environment. Since these are the objectives of almost all the commercial middleware products, a standard Corba compliant ORB is used since Corba supports object oriented modelling and allows distribution of the platform over a heterogeneous set of computers. Corba ORB establishes a client-server relationship between components. The client can transparently invoke an operation on a server object, which can either be on the same machine or across a network. Corba

architecture uses an Interface Definition Language (IDL) to describe the interfaces and an interface description defines the services of a component.

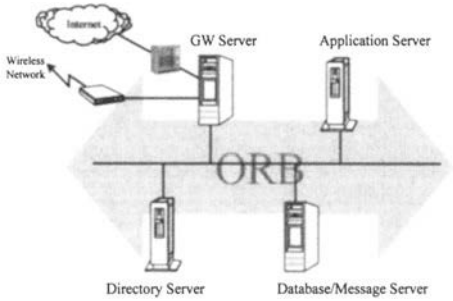

Figure 3. Distribution of UMF

The DPP also provides features, which support run time operations such as load balancing. Load balancing shares the load of the framework and the applications between several computer nodes.

By using a distributed platform such as Corba it is possible to integrate and still distribute the various components of the UMF over several network nodes. This kind of solution offers not only distribution benefits in general, but also benefits specific to UM and frameworks:

1. One of the principal drivers behind the framework concept is to provide an easy and cost effective means to increase the performance of the UM solution. Distribution of UMF and UM applications over several machines allows service providers to start with a simple configuration and a limited set of services without having to make huge investments in the UM solution.
2. The framework concept also enables customisation of a UM solution for specific environments. A small-scale framework can initially comprise only the essential functionality and a small set of services but may still be expanded to a higher performance system offering a richer set of complex services. This kind of expandability allows service providers to prepare for future messaging access methods such as WAP and UMTS.

4.1.2 Network/Protocol adapters

Network/Protocol adapters have two main functions. They enable different kinds of terminals to access UMF applications and they also provide communications resources to the applications. Each network adapter implements a protocol specific Corba interface. By using the Corba interfaces all of the other components, such as the applications running on the top of framework, can communicate with the adapters. A framework can contain one or more network adapters depending on the environment and the

needs of the service provider. The number of the adapters can also vary over time.

In the diagram below, the SMTP adapter is used to access the UM application. This application uses the SMS adapter to send a short message notification to the mobile phone of the receiver of the email.

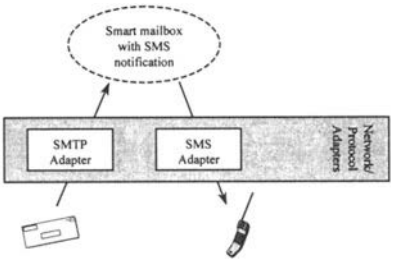

Figure 4. Network/Protocol adapter services

A UM application developer can freely combine the services of the adapters and create any type of sophisticated and advanced services, which may be required. Because new adapters can be easily plugged to the framework, the framework does not limit the imagination of the developer.

The following examples detail possible uses for the network/protocol adapter services. Note that only voice and SMS adapter is presented:

SMS adapter allows its clients to

- send short messages; and
- receive short messages.

Voice adapter allows its clients to

- receive a call;
- make an outbound call;
- detect DTMF tones;
- generate DTMF tones;
- play voice messages; and
- record a voice message.

A voice adapter with the capabilities presented above, enables developers to create, for example, sophisticated voice mail solutions.

4.1.3 Common service components

The main thrust behind the framework approach is that there are common needs among the components of the environment. For example, most of the UM applications need a place for messages and therefore there should be a common message store for the use of all the applications. This kind of general requirement can be solved once as a common UM service

component which provide either Corba interfaces or proprietary interfaces to the client components.

Most of the common services are involved in management since the same management needs apply to all the applications and the UM framework itself. By providing a suitable set of common service components to the use of the UM applications and other components, the framework can be managed as a whole. This way a person who manages an environment like described in this paper has a consistent view to the framework. If these management issues would be solved separately by all of the applications, the result would be unmanageable. Management issues of the UMF have been described in chapter 5.

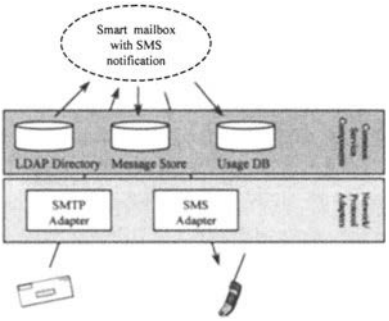

Figure 4. Example of using some common service components

In the example presented in figure 5 the common services used are the LDAP directory for user information, a common message store for the messages and a billing database for accounting. The application, which manages the mailbox can use the directory to retrieve the GSM number of the user who received the email. Since it is possible that other applications also use the same GSM number information, it is stored into a centralised LDAP data store so that it is available for all the UMF components. The retrieved GSM number can then be used to deliver a short message to the receiver of the mail. Because this kind of feature is not free, at least in this scenario, an accounting record is inserted to a common usage database which contains information referring to the usage of the services.

Figure 6. Common components of UMF

As a minimum the following common components are required in the UMF:

- Message conversion utilities for converting messages from one format to another (e.g. text-to-speech).
- A message store as a central repository for the messages.
- A directory, which is used to centralise the configuration information.
- A set of management tools.

Directory and management tools are addressed later in this document.

5. MANAGEMENT

A service framework such as UMF must take care of the management aspects. As mentioned earlier, there are two reasons for this:

- It is not practical to implement the management functionality individually by each UMF component.
- The use of the same management components enforces the whole UM environment into a single management policy.

The components of the framework should form a single Managed Information Base (MIB) for use by the system managers. The MIB must also be easily expandable since the configuration of the UMF will undoubtedly vary over the time.

A traditional approach to address management issues is the use of standard management models such as SNMP or OSI. There are also new ideas based on distributed object based management from the Directory Enabled Networking (DEN) working group and OMG. The DEN approach stores static management information into LDAP directory, but it does not address the management of transient information, which must be managed by the components themselves and accessed by using traditional methods such as SNMP protocol.

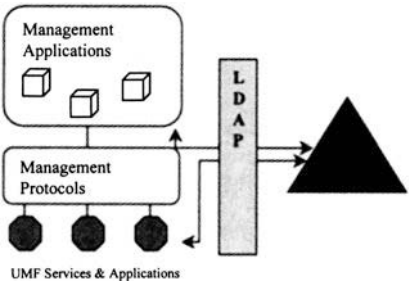

Figure 7. Management overview

Since the main aims are to provide a consistent view for management applications into the UMF MIB and a uniform way to store configuration data, then the DEN proposal offers the best solution. Static management information is stored in the LDAP directory provided by the framework instead of in the configuration files. In this manner a consistent view of the managed information is available without the use of application or component specific configuration files.

The use of LDAP also provides other benefits; the location of the UMF configuration information depends on the directory setup, which allows free placement and replication of this information. Placement and replication can be important issues when considering the performance and availability of the directory. LDAP also solves security and access control issues.

A single view to the transient information of the UMF components can be gained through the use of extensible SNMP agent technologies which enable a single SNMP master agent to be extended when new managed components are added to the framework. Extensibility is needed because SNMP architecture states that one physical machine contains only one SNMP agent which takes care of the management requests.

6. IMPLEMENTING UM FRAMEWORK WITH APPLICATIONS

The creation of an expandable UM framework for executing UM services takes time and requires resources yet, increasing competitive pressures demand that service providers deploy all new services at a fair price and in timely manner. This section provides a concrete roadmap for the development of UM framework, which meets existing and future requirements.

6.1 UM implementation strategy in nutshell

The roadmap for building a reusable framework together with application has been tested in real life. The following steps can be used:

1. Get the UMF model in place. Framework is a goal that the design team takes – no framework is being created accidentally.
2. Define the technical UM solution candidate(s). UM solution comprises the framework together with the UM applications.
3. Implement one of the solution candidates. Pick that solution candidate that best matches organisation's requirements.

6.2 Getting the UMF model in place

Clearly the design of the current interfaces provided by the various existing systems in the service provider environment provides a logical start to the project. The key issue here is to recognise commonly needed resources that will belong to the UM framework. The allocation of responsibilities appropriately among the existing and new software components will result in a robust model that guarantees UM framework expandability and UM service maintainability.

6.2.1 UM application functionality modelling

IT system, such as the UM framework with applications, is typically a black box that will deliver the required functionality after implementation. The following diagram illustrates the well-known email notification example. The UM system boundaries, the participating roles and necessary interactions are modelled. It is to be noted that the 'email notifier' function actually delivers short message to SMSC that is responsible for delivery of the message further to the email receiver.

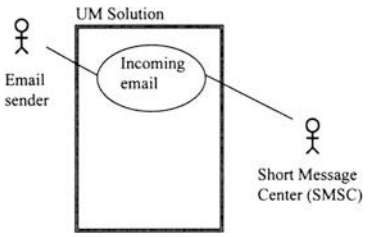

Figure 8. Email notify application requirement modelling

Now the UM application functionality has been modelled. The use case modelling is good in describing the needed functionality. However, no 'black box' functionality modelling technique promotes reuse of common components.

6.2.2 UM framework functionality modelling

The modelling technique shown in the previous chapter may result in correct email notification service functionality. However, it does not drive the implementation in the direction where the UM framework is clearly separated from the UM application functionality. To avoid the possible monolithic system implementation, a second modelling phase is needed for

the UM framework part. The following diagram illustrates the functionality being provided by the combination of UM applications and the reusable framework components.

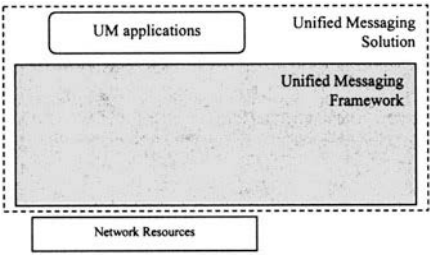

Figure 9. Division into framework and application part

This approach forces the solution to promote a separate interface for the framework part, which is needed for UM functionality expandability and maintainability reasons.

6.3 Define technical solution candidates

Technical solution candidates are being defined to fulfil the model that contains requirements for the UM services and the UM framework.

This section offers two scenarios for development of a new UM service. The first approach is based upon a tool centric start-up, observing the framework implementation from the service perspective, while the second start-up approach is framework component centric.

6.3.1 Tool centric start

Several vendors in the market are already offering UM solutions. These solutions normally provide a complete 'package' with set of ready-made UM services. Expansion of this kind of single vendor environment in accordance with the framework requirements described in this paper will enable swifter production of a UM service framework by comparison with the 'starting-from-scratch' approach.

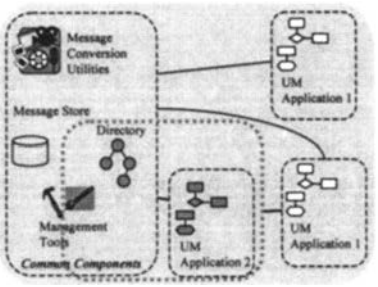

Figure 10. Selecting a UM tool and integrating it to the environment

Selecting a UM solution package requires a commitment to single vendor solution. This approach can involve limitations since telecoms equipment vendors tend to produce closed systems which can be extended only by themselves. Access to the solution's internal resources may limit the opportunities to integrate the tool with external components needed when implementing new services. The challenge for the service provider is to offer the services and service variations for the customers' changing needs while the environment may not provide the full implementation flexibility.

The best possibility would be to find a solution which can be easily tailored and extended by the service provider or a third-party software house.

6.3.2 Framework component centric start

The alternative method for building a framework is to start from the UM service needed and at the same time to produce a piece of the UM framework by abstracting common components from the services to be common framework components. The idea here is to enhance the UM framework with new common components while implementing subsequent UM services.

The standardisation of the management, directory and other components provides an opportunity to drive the implementation to more open direction. However, this approach will require a lot of integration and planning work to combine the UM framework components into a coherent single system, and consequently take longer to start the implementation of the actual UM services.

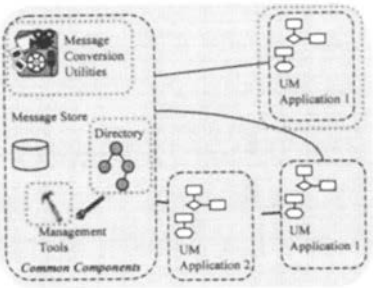

Figure 11. Developing UM framework while creating UM services

6.4 Implementing the solution

Since the core elements of the UM framework, such as the LDAP directory, are already available as commercial software packages, the main thrust of the process is the integration of these applications. Integration is typically implemented by utilising application programming interface (API's) that is a set of application's programmatic services. Because the framework presented here uses a distributed processing platform based on Corba as a communication method between the components, the integration must be implemented with its means. This implies that every component must provide Corba IDL interface for the use of other components.

7. SUMMARY

The unified messaging framework (UMF) approach offers fast time to market for new messaging services through utilisation of existing resources. Reuse of existing resources also is a cost efficient way to develop new services.

For gaining results both in short and long term the UM framework implementation strategy is being proposed in this paper. Efficient and cost-effective UM service development and maintenance require reusability and centralised management which are such important productivity factors that the way of achieving those features is not that important though.

Operators not offering value added services such as UM will have to continue to rely upon price differentiation. If the decision is to offer value added messaging services then the framework based UM solution makes it possible to build and manage messaging services efficiently and cost-effectively.