A flexible service-level accounting architecture for telecommunications

A thesis submitted to the
University of Dublin, Trinity College
for the degree of
Doctor of Philosophy.

Cliff Redmond,
Department of Computer Science,
Trinity College,
Dublin.

October 2000.

Declaration

I, the undersigned, declare that this work has not been previously submitted as an exercise for a degree at this or any other University, and that, unless otherwise stated, it is entirely my own work.

Cliff Redmond
October 2000.

Permission to lend or copy

I, the undersigned, agree that the Trinity College Library may lend or copy this thesis upon request.

Cliff Redmond
October 2000.

Acknowledgements

I would like to thank Vincent Wade for his support and patience, and Professor Jane Grimson for her encouragement and guidance throughout the writing of this thesis. There is little doubt that without Vinny this thesis could not exist, and that without Jane this thesis would not exist. I would also like to thank my parents, brother and sister for their encouragement and understanding, and also especially thank Maura for her love, encouragement and forbearance throughout the process.

Thanks are also due to the many people I have had the pleasure of working with, directly or indirectly, during this research. My colleagues in KDEG have been great a source of support and friendship, as have many others in the wider Department of Computer Science. Thanks are also due to those in the European project partner institutions and companies I have had the privilege of working with. Thanks to Andrew Nolan, Brian Cullen, John McKeown and Paul Doherty, the current inhabitants of the lab., and to those who have taken part in the recent exodus: Ben Jung, Gaye Stephens and Eoghan Felton. Thanks especially to Jesús Bisbal, who has been a great support as a fellow sufferer in the past year. Thanks are also due to Conor Power, John Fuller and Patrice Delorme – who was always there, even when he was back in France! I enjoyed working with Conor and John very much, as I have with Koen Daenen in Alcatel, Detlev Mattes in GMD Fokus, Alina Da Cruz and Chris Malbon in UCL, and Michéal Crotty in WIT; thanks for making research work such an enjoyable experience!

Abstract

The opening up of telecommunications markets has forced a differentiation between service and basic connectivity provision; many providers have accepted that new services will become their main source of income in the face of regulations that ensure that the market incumbent, and other providers with significant market power, provide unrestricted network access in the common-carrier model.

One basic requirement for a new service is that it is capable of, at least, recouping its costs. This is usually brought about by charging for service use, normally with a complementary need to account for this use. Thus far, standardisation for accounting has concentrated on mechanisms based on data collected from the resources used to support the communications channel at the network level. In the common model this data is then correlated and aggregated to produce a service transaction record which is in turn used as a basis for charges. The applicability of this to multi-service networks in an age of abundant bandwidth is questionable; the most successful of such multi-service networks, the Internet, depends on a packet based transport mechanism whose network level usage data are not easily related to specific users, or even specific services. Even with an agreed basis for charging, as is the case with telephony, the interpretation of network level usage data to produce charges has largely been declared out of scope of standardisation, and most usage data, although produced in a standardised way, is interpreted to produce charges in a service specific way.

This thesis argues that network level accounting measures are not appropriate when accounting for services provided on multi-service networks. It proposes a value, rather than cost, based pricing mechanism founded on economic pricing models for network industries. It then suggests an architecture to support this mechanism. This architecture promotes a clear separation between a service's operation and its accounting, enabling faster service deployment.

Table of contents

Figures

Tables

1 Introduction

1.1 Problem context

The deregulation of telecommunications markets the world over, starting in the United States, has lead to increased competition with consumer discrimination based on provider service offerings. Deregulation has forced a differentiation between service and basic connectivity provision; many providers have accepted that newer services will become their main source of income [McKinney98] in the face of regulations that ensure that the market incumbent, and other providers with significant market power, provide unrestricted network access in the common-carrier model (for example [90/388/EEC]).

One basic requirement for a new service is that it is capable of, at least, recouping its costs. The complementary need to account for service use remains even where costs may be covered indirectly, for example through advertising. Thus far, standardisation for accounting has concentrated on mechanisms based on data collected from the resources used to support the communications channel at the network level. In the common model this data is then correlated and aggregated to produce a service transaction record which is then used as a basis for charges [Q.82598].

In the case of a network largely optimised to support a single service, built using scarce resources with high running costs, as was historically the case with telephony, service charges based on the use of network level resources are an intuitive and apparently fair way of ensuring that users who use more, pay more. Their applicability to multi-service networks in an age of abundant bandwidth is questionable. The most successful of such multi-service networks, the Internet, depends on a packet based transport mechanism whose network level usage data are not easily related to specific users [Edell95], or even specific services (although there have been recent attempts at service 'level' differentiation, see section 2.3).

Even with an agreed basis for charging, as is the case with telephony, the interpretation of network level usage data to produce charges has largely been declared out of scope of standardisation, and most usage data, although produced in a standardised way, is interpreted to produce charges in a service specific way [X.74298]. Indeed the definition of the service depends on a different interpretation alone in some cases (e.g. called-party-pays or 'Free-phone' services based on number translation. This is discussed in the review of charging for IN services in section 3.4.3)

1.2 Approach

This thesis promotes an accounting framework, operating at the service layer, that can recognise and charge for the added value that new services offer over and above basic connectivity. It examines the price basis of telecommunications services and argues that an extensible, standardised, service accounting architecture will enable greater software re-use and thus facilitate rapid technological and market-led service innovation and deployment.

The architectural approach is demonstrated in its application to two service paradigms: video-conferencing services and information services, services which are usually seen to have divergent accounting requirements. It an extension of the telecommunications service architecture produced by the Telecommunications Information Networking Architecture Consortium (TINA-C), a market-led standardisation forum. The architecture embodies a separation between a service and the underlying network, a separation which endeavours to give TINA applications network technology independence. TINA has tried to increase the speed of new service creation and deployment by defining generic, reusable, service software and support components.

The proposed component-based service accounting architecture has at its heart a clear separation between a telecommunications service and accounting for that service. This separation required an extensible description of a service for accounting purposes and a clear interface between the accounted-for and the accounting services, incorporating a price feedback mechanism that can prove vital for information services, and useful for more traditional services.

1.3 Thesis

This thesis argues that network level accounting measures are not appropriate when accounting for services provided on multi-service networks. It proposes a value, rather than cost, based pricing mechanism founded on economic pricing models for network industries. It then suggests an architecture to support this mechanism. This architecture promotes a clear separation between a service's operation and its accounting, promoting re-use of accounting components for new services and thereby enabling faster service deployment. Accounting at the service level is based on a specialisation of the TINA service generic 'session' concept [TOCP95]. (Several 'session' specialisations are used to provide context for multi-party service interactions in TINA systems). The architecture ensures that the service independence of the accounting mechanism is maintained by enforcing a clear, yet extensible, definition of services for accounting purposes.

1.4 Contribution

This thesis outlines a common price basis for disparate telecommunication services. It then proposes a component based service accounting architecture that supports the pricing schemes implied, including service usage based, i.e. dynamic, price setting. The application of this architecture to two disparate service paradigms, video-conferencing services and content based services, is then described, as is the architecture's inherent ability to support real-time service accounting.

1.5 Structure of report

This thesis begins, in the next chapter, with an examination of telecommunications pricing models. Chapter two also outlines requirements that the implementation of these models impose on accounting systems, particularly the charging mechanisms. Chapter three considers the development of software defined telecommunications services, then examines the ability of several important telecommunications software architectures to support a generic re-usable accounting mechanism, before examining the specific provisions made for accounting in these architectures. Chapter four outlines the primary design influences and requirements for a service accounting mechanism, and sketches an architecture to support these requirements. Chapter five outlines two successive and incremental renderings of this architecture, for four separate services, over the course of two research projects. Chapter six evaluates the architecture, its requirements and implementation, and summarises its contribution. Chapter seven summarises the thesis, outlines limitations in the approach and presents conclusions, then offers suggestions for further work based on this research.

2 Pricing telecommunications services

2.1 Introduction

Telecommunications services have long been priced based on the scarcity of expensively run resources and the belief in the existence of a ‘natural’ monopoly for their provision [Fowler86]. Rapid and continuous technological progress has consistently challenged these assumptions which are the basis of economic models used to help regulate the industry. Regulation will nevertheless be seen to be necessary for some time; it is felt that telecommunications have become too vital to the health of nations to be left to the whim of the markets [97/51/EC]. While it is debatable whether ‘natural’ monopolies do, or did ever, exist in telecommunications [Fowler86], [Perez94], state sanctioned oligopolies are still being created; one need only regard how recent exorbitant wireless licence fees act to restrict new market entrants. (Models for oligopoly pricing are similar to those for monopoly pricing [Mitchell91]). The price levels and structures of telecommunications will be based on economic models that may or may not represent reality for the foreseeable future; in effect they dictate reality. Accounting systems must be capable of representing and interpreting these structures to price services appropriately.

The pricing for newer ‘content’ based telecommunications services can justifiably be based on the existence of a monopoly. Owning original content is a true ‘natural’ monopoly; the holder is the only one who can sell it. The original copy represents the same high initial costs (*fixed* or *sunken* costs) that first dictated monopoly pricing in telecommunications. The low costs for subsequent copies (*marginal* costs) reflect the low costs of telephone calls once the initial infrastructure is in place (e.g. an electronic exchange supporting the maximum number of calls has the same operating costs as one supporting none). However, the ease with which information in digital form can be replicated has further implications for role of time in pricing and puts additional requirements on accounting systems.

This chapter outlines historical and contemporary methods of pricing telecommunications services and the requirements they impose on accounting systems. The next section outlines the development of the telephone industry and the economic models used to regulate the price of service. Section 2.3 outlines the impact of the Internet on service provision and pricing, while section 2.4 describes the ‘economics of information’ and its price basis. The requirements for accounting systems are summarised in section 2.5.

2.2 The development of the telephone industry

Until relatively recently most telecommunications service providers, or network operators, whether private or public, provided telephony services on monopoly basis. This section discusses the development of these monopolies and the regulated pricing regimes which applied to them and apply to the oligopolies which replaced them.

2.2.1 Monopoly

Alexander Graham Bell established the ‘American Bell Telephone Company’ in 1877, the year after he filed his initial telephone patent. This patent, upheld through a series of legal challenges on would-be competitors [Farley], granted his company monopoly status until it expired in 1893 [Perez94]. The company, by then called ‘The American Telephone and Telegraph company’ (AT&T), then overcame its many new competitors through various means (some, like limiting interconnection, would be regarded as anti-competitive now) until regulation was forced upon it (although not altogether reluctantly) in an agreement with the U.S. Department of Justice in 1913 (the ‘Kingsbury’ agreement [Perez94]).

The agreement was not opposed on economic grounds, as at the time, many economists felt that regulation was inevitable: telephony was seen as a ‘natural monopoly’, i.e. competition would ultimately prove wasteful as a single supplier could provide better service at lower costs than a number of competing suppliers [Fowler86]. The reasoning behind this included the assertion that subscribers would eventually exclusively join the largest network. This assertion is now attracting considerable scrutiny in the light of evidence that individuals did not necessarily “swing to the larger system”: the competing telephone companies tended to cater to different market segments and subscribers often chose to join more than one network [Perez94].

Accepting the ‘natural monopoly’ hypothesis, other countries, particularly in Europe, opted to provide telephony service as a state monopoly, rendering separate regulatory bodies unnecessary. Typical of such state monopolies was the British General Post Office, which in 1912 extended its existing monopoly over posts and telegraphs to become the monopoly provider of telephone services in the United Kingdom [Littlechild79], [BThistory]. It accomplished this by buying independent telephone companies, with the exception of the Hull municipal system which remained independent. In the absence of competition such state monopolies relied on economic models to dictate a level of service and the pricing of that service.

Initially, two telephony services were available; local and long distance service. Local service was provided to subscribers using the same exchange, where no other facilities needed to be used to provision a call. Long distance service involved the use of trunk, or toll, lines accessed by local exchanges through trunk exchanges which existed at another level in the switching hierarchy.

Although they shared common plant, i.e. the telephone and local exchange, local and long distance were regarded as separate services: they had separate price schedules and applied to different market segments. At the outset both these services were labour intensive with significant running costs, for both labour and plant. Exchanges required paid switch board operators, and the facilities providing long distance service were expensive to run with power hungry, thus costly, valve based amplifiers in need of constant replacement (the transistor was invented to overcome these high costs). As technology improved, operators were replaced by electro-mechanical switches, but these still occasioned maintenance costs.

In economic terms, the sharing of facilities means that these telephone services have joint costs. The difficulty in the allocation of joint costs between separate local and long distance telephone companies was (and is!) the subject of considerable debate; some have regarded the 'separations procedures' which apply in these cases as arbitrary [Littlechild79]. State monopolies avoided this problem by their nature.

The high running costs meant that early telephone services had a significant variable cost for each call, i.e. each call cost the system money. Economic theory maintains that for industries where marginal cost[*] *increases* with output, optimum allocation of resources is achieved by selling a product at its short run[†] marginal cost (where costs include 'normal' profit) [Yordon84], [Stanlake95]. Local telephony service was largely viewed as an increasing-cost activity in the first half of this century, but this view's "validity became dubious with the spread of mechanical switching" [Yordon84] with its low variable costs. Since then variable costs have continued to decrease; switch operating costs are now nearly independent of load.

---

[*] "The word 'Marginal' is a used to designate the borderline unit; it is the economists' quaint term for what mathematicians call a first derivative" [Yordon84]. Marginal cost is the cost of an additional unit (i.e. the total cost (fixed & variable) of n units - the total cost of n-1 units).

[†] 'short run' means in the period between significant capital expenditures (e.g. for switches). Such expenditure imposes a 'fixed cost' per unit in the short term but this can change in the long term (e.g. with improving technology). Thus a 'short run marginal cost' is the cost of a marginal unit at present capacity.

With the advent of automatic switching the investment per subscriber began to fall as more people subscribed. The actual investment needed continues to fall as switching technologies have become cheaper and more reliable [Littlechild79], [Yordon84]. This leads to a decline in both short and long run marginal costs; the share of the fixed cost decreases with each subscriber and the fixed costs themselves decrease with technological progress.

The overall decline in marginal costs constitutes what economists variously call an ‘externality’, a ‘network externality’, a ‘network effect’ or a ‘demand side economy of scale’ [Yordon84], [Shapiro99]. Such externalities mean that the traditional economic solution, where prices are set to apparent marginal costs, “would probably result in deficits for the telephone companies” [Yordon84], because fixed costs could not be covered.

Such atypical economic conditions lead to considerable problems in choosing the right rate, or tariff, structures and in setting their levels. (Although it has been noted that in the public monopoly paradigm adopted by many European countries there was the consideration that any supernormal profits could be used as an alternative source of government revenue! [Cairncross95]). A dedicated communications regulator, the ‘Federal Communications Commission’ (FCC) was established to oversee the private monopoly regime in the United States under the 1934 Telecommunications Act. This act dictated ‘Universal Service’ in its preamble. This was an inclusive social policy mandating a single price for connection that also happened to suit the monopolist; while isolated and rural communities could get relatively cheap telephony service, higher density urban populations (with significantly less overhead per subscriber) got it at the same price. This concept still affects pricing policy although its applicability in a non monopoly market is dubious [Browning94], [Mueller97].

2.2.2 Regulation

The goals of the regulator dictated the policies it used to control prices. Similar goals and policies still apply in many newly ‘deregulated’ markets and are used in framing tariffs. The goals of FCC regulation were to be fairness and efficiency, where fairness “encompasses the fundamental goals of reasonable rates, the absence of unjust discrimination and universal service" [Fowler86]. Economic efficiency can be defined in terms of 'welfare maximisation' [Mitchell91], where welfare is a function of the utilities of all the members of society*.

Regulators sought a theoretical basis for rate structures and levels, albeit while overall levels could be largely dictated by executive fiat. Unable to use the optimal price basis (i.e. 'price at marginal cost') because of the declining marginal cost nature of the industry they sought other ways to price service to allow operators to make profits at controlled rates of return.

Where service demand can be correctly anticipated, the 'peak-load' pricing rule can be used [Mitchell91]. In off peak periods the price can be set equal to short run marginal cost (i.e. near zero for local exchange service) while at peak load the price is set equal to long run marginal cost (i.e. a unit cost for expanding capacity in the long term is added). While theoretically viable, in practice this method of pricing would lead to a "shifting peak" which would make the pricing rule inconsistent because the peak-load price would be significantly more than the off peak price. This problem can be mitigated by spreading the cost of expanding capacity over many peaks where prices are set high enough (but not too high) to constrain demand to just equal the capacity available [Mitchell91].

There are numerous problems with peak-load pricing, including accurately forecasting demand. Inaccurate forecasting can lead either to excess capacity resulting in operator losses or 'non-price rationing' (i.e. busy tone) and subscriber frustration! Feasible tariffs were also limited to a "few prices per day … so a period with a single price may include a range of demand levels". A possible solution to irregular demand was the notion of 'real-time' pricing, but in [Mitchell91] the authors poured scorn on the idea of such 'spot' pricing because "under regulation, tariffs will have to be steady and therefore based on anticipated long-run patterns of demand" (an exquisitely circular argument). (It is worth noting that the Internet now

---

* Utility is the basis of neo-classical economics' theory of value, and although derided as "a metaphysical concept of impregnable circularity" ("utility is the quality in commodities that makes individuals want to buy them, and the fact that individuals want to buy commodities shows that they have utility"), brought mathematics to economics and "seemed to promise a new dawn for economics as truly scientific subject" [Robinson62].

The concept of welfare itself has been attacked by members of the 'Austrian school' of economics: "The truth, as seen by the Austrians, is that economic welfare – consisting as it does of nothing but the subjective sense of well-being of separate individuals – displays an interpersonal incommensurability which simply defies aggregation." [Kirzner81].

supports several ‘bandwidth exchanges’ supporting just such spot markets for wholesale capacity [BandX97], [RateXchange98]).

Other problems included the basis of peak-load pricing, the notion of ‘marginal cost’ itself. According to [Littlechild79] “the injunction to set prices equal to marginal cost does not specify in an objective way the price to be set. The calculation of the cost is necessarily left to the discretion of the industry… The point is that the injunction to set price equal to marginal cost is an empty one: it amounts to no more than an injunction to set prices in a ‘reasonable’ way.”

Telecommunications regulators turned to quasi-optimal pricing, or Ramsey pricing. This method of pricing “allows the firm sufficient use of monopoly power to meet its revenue requirement.” [Wilson93]. Ramsey pricing seeks to maximise welfare and can be expressed non-mathematically as the “inverse elasticity” rule:

> “All relative deviations of prices from marginal costs should be inversely proportional to the corresponding demand elasticities*.” [Mitchell91]

This basically states that products that are deemed necessary and that have no near substitutes can be priced higher for certain market segments (this is also used as the basis of some taxes!). A measure of welfare is “the money value of consumers’ utility minus the money cost of production”, which “assumes that a dollar has equal value to rich and poor alike” [Yordon84]. This made Ramsey prices “hard to implement politically if they imply a price structure that is quite different from the status quo” [Mitchell91]. Taking account of this, constraints, including ‘Pareto optimality’†, were then used to “ensure customers’ and regulators’ acceptance of Ramsey pricing as an improvement over an existing uniform price schedule.” The effect of these constraints is “usually to put a cap on the prices charged for small purchases” [Wilson93].

---

[*] elasticity is the ratio of the relative change in quantity demanded to the relative change in price. In effect elastic demand is characterised by a product with close substitutes or potential for delayed consumption.

[†] ‘Pareto optimality’ is brought about when one individual’s utility can rise without a consequent fall in another’s.

Ramsey pricing is a form of 'value-of-service' or discriminatory[*] pricing, in that "if it is possible to segregate users who will not be deterred by a high fee, they can be made to bear the financial burden of the construction cost without curtailing the use of the facility … more generally it is a system where cost of service establishes the minimum charge and a variable mark-up is added to collect additional revenue from those who value the service highly" [Yordon84].

Ramsey prices can be improved upon by selling additional units nearer to their marginal cost; this is called 'non-linear pricing' or 'non-uniform' pricing [Mitchell91], [Wilson93]. This is another form of discriminatory pricing; different units of the same service are still sold at different prices, but this time to the same consumer. Such prices are "widely applied in telecommunications" although "their derivation requires more detailed information about individual demands" [Mitchell91].

The simplest non-linear tariffs are 'two-part' where there is an entry fee, normally fixed, and a fee for each additional unit, perhaps the marginal price [Mitchell91]. Such tariffs can be inefficient "if the fixed fee excludes some customers from purchasing" the main effect being "a reduction in market penetration in exchange for higher fixed fees collected from the market remaining." [Wilson93]. Any commitment to universal service would therefore appear to have some bearing on the level of any fixed fee. Other non-linear tariffs are 'multi-part' and 'smooth' tariffs which allow smoother price changes with the quantity purchased and can iron out "distortion" caused by applying a relatively simple 'two-part' tariff [Mitchell91]. Figure 2.1 illustrates several tariff types.

In common with unaltered Ramsey tariffs, non-linear tariffs are thought best applied where the seller has monopoly power, although "feasible in markets that are imperfectly competitive" like oligopolistic competition [Wilson93]. The Ramsey inverse elasticity rule (as stated above) also applies to non-linear pricing [Mitchell91].

The monopoly status of AT&T was continually challenged throughout the early part of the century. Competition gradually resulted; first based on acoustically coupled interconnection, then in the equipment market. With electrical interconnection a legal possibility, competition soon followed in the long distance market. The FCC granted licences to Microwave

---

[*] [Yordon84] states that "no normative significance should attached to the term 'discrimination' in the context of public utility economics; one must make an effort to avoid the common language connotation of wrong doing."

Communications Incorporated (MCI) to be a carriers in the US long distance market, direct competition for AT&T's 'Long Lines' service. Cross-subsidisation of local service was becoming increasingly untenable, and 'usage-sensitive', i.e. metered, local calls were introduced to replace flat rates by a number of regional telephone companies [Garfinkel75], [Carlton75]. Ironically, this 'message rate service' replaced flat rates for local calls at a time when extremely efficient electronic switching systems were coming to the fore.

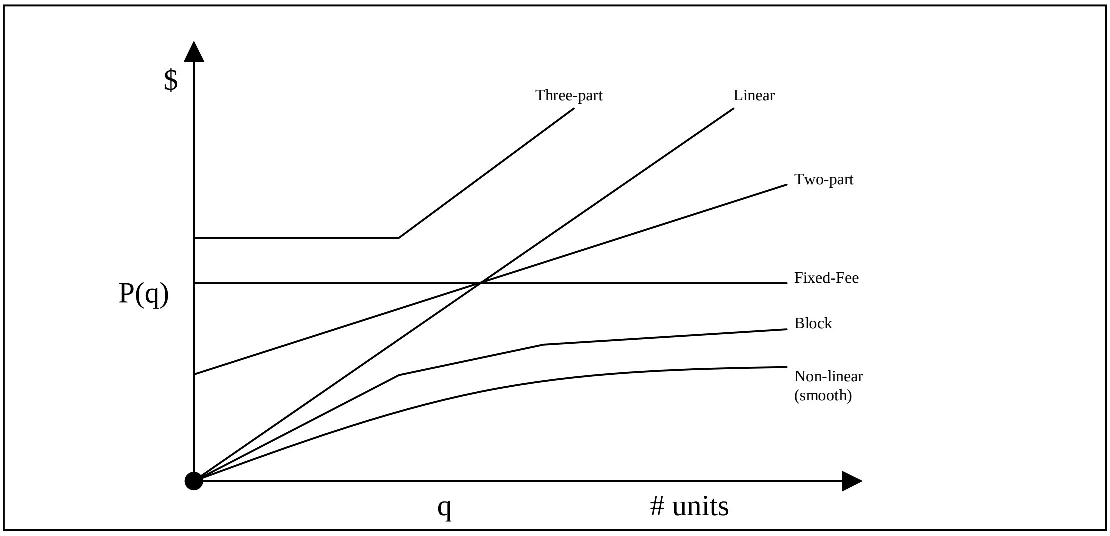


Figure 2.1, An illustration of tariff types (Fig 1.1 from [Wilson93])

2.2.3 Oligopoly and oligopoly regulation

Competition came to the United States' telecommunications market with a gradual recognition that "governmentally bestowed monopoly … creates strong incentives for overpricing and reduced output of the monopoly services" and that it had "extracted significant efficiency costs in resource allocation: distorting investment decisions, limiting private incentive to innovate with new technology, and worse, affirmatively discouraging innovation that would render obsolete vast amounts of embedded equipment that is included in the rate base." [Fowler86]. Similar inefficiencies were to be seen in the state run monopolies of Europe. AT&T was broken up into separate local and national service providers in 1984, its monopoly in long distance telephony having been completely removed in 1980 [Yordon84].

Writing in 1986, the Chairman of the FCC stated that "the past thirty years have made it clear that the public utility paradigm does not apply, and perhaps never should have been applied to the entire telecommunications industry. The persistence of potential competitors, plus the development of new technologies, has effectively undermined the notion that outside companies can never become effective competitors of the telephone companies", suggesting that that "the competitive industry paradigm" might also be the best "long-term" model for local telephony with the (perhaps idealistic) belief that competition would drive prices so low that "no need will exist for subsidies for telephone service to any American consumer". [Fowler86].

Defenders of the public monopoly paradigm, popular in Europe, ceded that "the regulation of profits, or constraints on the pricing system adopted, may have the effect of reducing the financial incentive to provide a telephone system free of congestion" before suggesting that public monopoly reduced financial constraints! [Littlechild79]. This view was expressed before the renaissance of orthodox neo-classical economics in the 1980's*. Observing the successes of competition in the United States, and realising that Britain needed to be competitive in a global market (perhaps especially because it was English speaking), the British government passed the 1984 Telecommunications Act, privatising its public telecommunications provider while abolishing its exclusive privilege to run telecommunications systems and establishing a regulatory framework to "safeguard the workings of competition" [BThistory].

Britain was at the vanguard of market 'liberalisation' in European telecommunications. Competition was introduced in other European markets under the aegis of the European Commission, first in the telecommunications equipment market under directive 88/301/EEC and then some telecommunications service markets under directive 90/388/EEC and council directive 90/387/EEC ('Open Network Provision' (ONP)). Competition was extended to voice services under directive (95/62/EC) [EUTP99]. Competition was accepted reluctantly by some member states, including Ireland, which sought, and obtained a derogation to delay its introduction by two years [97/114/EC]. In reality the introduction of regulated competition, perversely called 'deregulation', was achieved a year ahead of schedule in the Irish market.

Regulation was necessary for a number of reasons; without it, new operators could 'cream skim', i.e. they could take business away from the former monopolist (now called an 'incumbent operator' or 'incumbent') in profitable market segments without serving the less profitable segments [Yordon84]. Regulation was also thought necessary to prevent the incumbent from abusing its erstwhile dominant market position by temporarily undercutting

---

* An eminent Cambridge economist remarked that "economics itself has always been partially a vehicle for the ruling ideology of each period as well as partly a method of scientific investigation" [Robinson62].

any competitors to force them out of the market, and to enforce ‘Universal Service’ obligations, usually befalling the incumbent.

Monopoly regulators, where they existed, had previously examined potential competition where “the emphasis [was] on the existence and properties of tariffs that exclude inefficient [market] entry”, [Mitchell91], i.e. to dissuade new entrants. With the introduction of competition, both newly formed regulators and established former monopoly regulators needed to take this new environment into account when framing tariff structures.

Economists adapted oligopoly, or dominant firm, models interpreting an oligopoly as “a ‘monopoly’ with adjusted elasticities”, where “a small firm in a fiercely competitive market can be viewed as a monopolist in a highly elastic market.” [Mitchell91]. Thus second best pricing policies, like Ramsey pricing, were still deemed applicable, particularly because “economies of scale can prevail at the outputs produced by some of the oligopolists in equilibrium”. In [Mitchell91], the authors demonstrated that profit maximising oligopoly tariffs had a similar structure to welfare maximising tariffs, but that the price level would differ; because of this they expected that “under profit maximisation the tariff innovations will be quite similar to those required under welfare maximisation”.

Recent European telecommunications legislation has tended to dictate cost based tariffs, for example directive 97/51/EC, an amendment to the ONP directive, states:

> “…tariffs must be based on objective criteria and, until such time as competition becomes effective in keeping down prices for users, must in principle be cost oriented, on the understanding that the fixing of the actual tariff level will continue to be the province of national legislation and is not the subject of open network provisions conditions. Where an organization no longer has significant market power in the relevant market, the requirement for cost orientation may be set aside by the competent national regulatory authority” [97/51/EC].

Such legislation is intended to reduce service cross-subsidisation, which, in effect, mandates the allocation of joint costs between different services. Traditionally, this was done using ‘fully distributed cost pricing’ where joint costs were allocated by applying an agreed formula. These ‘separations procedures’ were regarded as arbitrary [Littlechild79], were held to have “limited fairness properties” and were “quite fragile in situations of competitive entry” [Mitchell91]. Access charges (or interconnect charges), as they are now known, are now predominantly based on ‘forward looking long-run incremental costs’ which is a form of long run marginal cost pricing[†] which takes into account such factors as depreciation and technological progress [Laffont00].

2.3 The Internet

The history of the early Internet is well documented (e.g. [RFC1642]), as are the technical differences between its underlying network and that underlying conventional telecommunications services (e.g. [RFC1958]). One of the most important aspects of the Internet is the fact that its packet switched transport mechanism enables multiple services to share the same telecommunications infrastructure easily. The importance of a multi-service network was recognised by traditional telecommunications service providers, but their attempts at an 'Integrated Services Digital Network' has so far met with little market success [Anania97]. Perhaps one reason is that the Internet also significantly eases service creation; Internet services are by their nature software defined and network technology independent, with services running between edge nodes of the network [Isenberg98]. Traditional telecommunications service providers' attempts to enable easier service creation are documented in the next chapter.

At the moment the majority of pricing schemes in use for the Internet are based on 'connection pricing' or 'flat-rate' pricing, where a user, or an organisation representing a number of users, pays a fixed fee for a fixed bandwidth connection. These schemes suffer limitations which many consider a 'tragedy of the commons'[*]. In economic terms a negative externality arises because a marginal packet sent by a user imposes a cost on all other users because the resources are then not available to them: "without an incentive to economise on usage, congestion can become quite serious" [Mackie-Mason97a]. This has led to a number of pricing schemes which internalise[‡] this negative externality, examples include various general usage sensitive, congestion pricing and/or priority pricing schemes [Edell95], [Brownlee97], [Mackie-Mason95a], [Mackie-Mason97b], [Gupta97].

The 'Smart Market' responsive pricing approach mooted in [Mackie-Mason97b], requires that an already congested router devote some of its resources to holding an auction based on

---

[†] Long run marginal cost pricing can take into account the 'lumpy' (i.e. infrequent but capital intensive) nature of investment in new plant.

[*] This is the common moniker for the concept, but as noted in [Ridley97] the originator of the idea (G. Hardin) now accepts that it would be more correctly referred to as 'the tragedy of the unmanaged commons'.

[‡] Internalising an external cost means making it a part of the price.

willingness to pay indications in each packet header. Any such non-edge based congestion accounting mechanism must be very efficient [Schnizlein98]. This scheme means that the user must be willing to make price based decisions in a ‘closed loop’ price feedback scheme. Each packet must also be directly attributable to a particular account. The additional overheads have lead one critic to remark that “the pricing solutions [are] getting more intricate than the networked interconnection itself” , while extolling the virtues of flat-rate pricing [Anania97]. However, the long-term viability of flat-rate pricing for Internet access has been questioned [Pioneer97], [CNET98] while congestion based prices can generate revenue for capacity expansion [Mackie-Mason95a].

Priority pricing schemes (e.g. [Gupta97]) benefit from architectural support for a simple packet priority differentiation mechanism in the Internet protocol (IPv4) which can be used to provide explicit differentiated service classes, unlike the ‘Smart Market’ approach which can only offer relative priorities. Such differentiated service classes also form the technical basis for ‘Paris Metro Pricing’ (PMP) [Odlyzko97], [Odlyzko98a] which the author explains “is about as simple as that of any usage-sensitive pricing scheme that has been proposed for the Internet … the additional complexity it would introduce is minimal, and appears inevitable, since usage-sensitive pricing appears inevitable” [Odlyzko97]. PMP intends to “reduce the traffic management task by inducing users to separate themselves into classes with different requirements”, each with a different capacity and priced differentially. This, the author hoped would “permit dispensing with measures such as RSVP and their complexity, and go back to the simpler model of the traditional Internet”.

Other, usage based, if not usage sensitive, network level charging schemes include one based on “expected capacity” [Clark96], [Clark97]. Here the idea of an expectation, as opposed to a guarantee, of service is used to provide a compromise between the current ‘best-effort’ service and guaranteed service [Clark96]. Users negotiate a traffic contract within an expected capacity profile and packets are marked as being within or without the profile, in a scheme similar to token schemes used for ATM Variable Bit Rate service [Bodamer98]. The pricing scheme is then used to ensure that asking for a better service has a higher price.

Usage sensitive schemes built on network level parameters have been criticised for a number of reasons. Some say such schemes often fail to recognise that different services have different performance objectives and that, for example, delay as well as through-put ought to be considered [Wang96], for example with voice services. Network-usage based schemes are objected to in general in [Anania97]: “It is impossible to simply count bits and come to any conclusion about the relationship between tariff structure and value”. This was reduced to the pithy maxim “not all bits are equal” in [Huber97]. In general network level charging schemes also fail to address some fundamental cost attribution issues that exist at the service layer, for instance those of multi-cast based services and whether the receiver or sender should pay for services like the World Wide Web [Clark96]. This has lead to attempts at mixed service level and network level charging [FTD398]. Very few of these charging schemes have been applied in practice, due to user and official antipathy as much as any technical difficulty in their application. Given the positive externality of the benefit of additional Internet users, flat rate pricing may continue for the medium term, perhaps with indirect government encouragement, in an attempt to stimulate more users to join.

Services on the Internet are not regulated. Indeed in the U.S., ISPs are exempt from Universal Service obligations under the 1996 Telecommunications Act. Without regulation, service providers, which will be software providers, can set prices for new services to encourage demand side economies of scale (e.g. free applications). In the absence of other sources of income, these providers could receive a stream of income by building-in functionality that supports service based accounting; this is similar to Cox’s scheme for ‘meterware’ [Cox96] except the charges would be for communication, not software functionality. The infrastructure provider could then receive an indirect payment, if they were not already owned by the service provider; such convergence has already started.

To be successful, such services should provide enough advantages to encourage users to continue to use them in the presence of a suitable substitute. The advantages could include ease of use and ease of discovery, but in the medium term quality of service guarantees will probably be important, if not in the long term [Fishburn98]. If persuasion does not work, it likely that subtle ‘lock-in’ techniques like making inter-communication with substitutes difficult, or more blatant ones, equivalent to refusing inter-communication, will evolve. If the service provider abuses their position of dominance wholesale service migration is always an option but it is likely that in such service markets, as in all software markets, there will always be significant ‘first mover’ advantages[*] [Shapiro99]. In such a scenario, regulation may make a comeback [MitchellR97].

---

[*] If conventional wisdom, i.e. that of the ‘New Economy’, is to be believed.

2.4 The information economy

Information is the basis of many new services available on the Internet. The growing importance of the ‘information economy’ has made the traditional economic debate revolving around information’s role as a private or public good all the more critical. This debate considers how to balance the rights of creators and the people that use their creations. As information has become divorced from physical media, its property of being easily copied and communicated has caused considerable problems to content providers, the creators of information. Network delivered information exhibits a “first copy” problem, in that once the costs for the first copy are sunk, additional copies can be produced at almost zero marginal cost [Shapiro99]. Recognising this, Barlow claimed that “almost everything we think we know about intellectual property is wrong”, proposing that a predictable reward for creation will be unlikely in the future: “Humanity now seems bent on creating a world economy primarily based on goods that take no material form. In doing so, we may be eliminating any predictable connection between creators and fair reward for the utility or pleasure others may find in their works” [Barlow94].

Dyson suggests that “the trick is not to control copies of your work but instead a relationship with the customers – subscriptions or membership” [Dyson97]. She notes that “the only fungible, unreplicable value in the new economy will be people’s presence, time, and attention; to sell that presence, time and attention outside their own community, creators will have to give away content for free”. (This mirrors an economic theory based on the finiteness of a person’s attention [Goldhaber96]). Dyson suggests a service based approach: “aside form a few leaders who manage to sell brand-name content widely and cheaply, the most promising business in the Net will be services and processes.”

In [Shapiro99], the authors echo this sentiment. It is asserted that there are only two sustainable structures in an information market; the dominant firm model and the differentiated product model. The dominant firm model resembles that of a monopoly service provider. Economies of scale gives the firm a cost advantage over competitors. In the differentiated product model, different varieties of the same good are produced and sold. If it is possible to produce personalised versions of information, then price personalisation is also possible; this is called “first degree price discrimination”, or one-to-one marketing [Shaprio99]. Other degrees of price discrimination are also possible, with third degree, or group pricing being the most common.

First degree price discrimination is possible without personalisation; different users could have a different willingness to pay for the same information, but as pointed out in [Huber93], “few customers will volunteer to be on the pricey end of price discrimination[!]” The author points out that it is illegal under U.S. Law, but asserts that “there’s no way to avoid usage-sensitive pricing in the info-biz. So law or no law, you find some way to measure just how much each consumer really likes your product and charge accordingly,” before brazenly predicting change: “Information just doesn’t obey the ordinary laws of economics, so the people who sell it can’t obey ordinary anti-trust laws. Judges had better get used to that. What we’re talking about here is the future of our entire economy”.

2.5 Summary and recommendations

The price basis of telecommunications is changing. Regulation is forcing network providers to open their networks to other service providers; the providers themselves accept that new services will become their main source of income [McKinney98]. At the same time, if true competition occurs, cross-subsidisation of services is unlikely to be viable in the long term. The contemporaneous move towards multi-service networks with edge-defined services leaves network operators finding it increasingly difficult to extract benefit from the sunken costs of network provision. Operators need a means to extract value from the services running on their networks, but thus far there have been few suggestions. Their long term viability seems to be increasingly dependent on content based services, witnessed by recent mergers between the two. Pricing for content, rather than network provision, looks increasingly likely in this scenario.

The requirements for a telecommunications accounting system coming from the analysis in this chapter, include; flexibility of price setting in the face of regulatory reform, with an allowance for price caps to prevent over and under pricing; the ability to model the range of tariffs identified in section 2.2.2, including non-linear tariffs[*]; the ability to give price based feedback to users and the ability to have per-user tariffs.

---

[*] It is noted in [Wilson93] that “… a non-linear tariff is costly or impractical to implement in most applications, so nonlinear tariffs are approximated by a menu with several optional two part tariffs, a single tariff with several linear segements, or a block defining price schedule with several steps”.

3 Telecommunications services

3.1 Introduction

In the previous chapter some of the economic and pricing aspects of network based service provisioning were examined; this chapter examines telecommunications services, and accounting for those services, as seen from the telecommunication service engineer's viewpoint. It examines the nature of telecommunications, revealing the move towards software defined services and examines the ability of several telecommunications software architectures to support a generic re-usable accounting mechanism, before examining the specific provisions made for accounting in these architectures.

The next section traces the development of software based telecommunications services, giving a brief history of software in telecommunications. Section 3.3 then describes software architectures underlying many current services and one which may underlie future services. It evaluates the possibility of each defining a generic and re-usable means of accounting for telecommunication services. The first subsection discusses IN, the 'Intelligent Network'. An analysis shows that, overall, the IN initiative was limited by technology and failed to sufficiently consider operations support issues for new services. The next subsection discusses an operations architecture, TMN, the 'Telecommunications Management Network' which, amongst other aims, intended to standardise management (operations) systems for telecommunications; it is shown that while the architecture was moderately successful in supporting standardised and well defined services, it failed to support new services or service creation. The last subsection discusses TINA, the 'Telecommunications Information Networking Architecture', an architecture which combined management and service issues. It is analysed with reference to the perceived shortcomings of the previous architectures and its use as the architecture underlying this research is explained.

Section 3.4 describes telecommunications accounting in general before describing the support for accounting in each of the software architectures described in the previous section. The analysis demonstrates that the general criticisms of both IN and TMN apply to their support for accounting and also highlights several shortcomings in the TINA accounting models. Section 3.5 contains a brief summary of the models and their accounting and billing facilities, highlighting the general requirements for accounting systems.

It is worth outlining the meaning of the term ‘telecommunications service’ for the purposes of this research. The Telecommunications Information Networking Architecture Consortium (TINA-C or TINA Consortium) proposes this definition of a service:

> **Service** : A meaningful set of capabilities provided by an existing or intended set of systems to all who utilize it. [TINAGOT97].

The scope of this quite general definition can then be reduced so that a telecommunications service is a ‘meaningful set of capabilities’ provided to the users of a telecommunications network. In economic terms different services correspond to different markets [Mitchell91], so while the ‘meaningful set of capabilities’ might be similar, they may be priced differently for each market. This research therefore uses a combination of these definitions when considering services for accounting purposes: a service is ‘a meaningful set of capabilities’ provided to a specific market which consists of a subset of telecommunications network users.

It has been asserted that the history of telecommunications service engineering has revealed the need to define services independently of the underlying communications technology (i.e. ‘the network’) in a manner analogous to the hardware independence compilers afforded early software engineers [MaA94a]. Many industry groups and standards bodies can be seen to have acted in the light of this demand, even if it was never explicitly stated in their documentation (although it has been by some [TReq95]). In the following sections it can be seen that the architectures generally differ by the degree to which they recognise, acknowledge or hold to this demand. The following section outlines the gradual development of software based telecommunications services.

3.2 Development of telecommunications service software

The introduction of computer technology into telecommunications started in the 1960’s when analogue switching systems had their common control facilities replaced by a ‘control program’ (common control refers to the use of a pool of registers to facilitate number translation and hence routing). These ‘stored program control’ switches initially replaced facilities implemented using electromechanical relays. Later as fully electronic switching systems based on digital switching techniques became available [Ithell89], it became easier to extend and amend the ‘control program’ to provide services over and above basic telephony to telecommunications subscribers. The replacement of electromechanical hardware with programmable controllers thus facilitated easier introduction of new services, i.e. hardware modifications were no longer necessary.

The advent of centralised control put new requirements on the reliability of the control system (both hardware and software), with at least duplication of facilities to provide the redundancy necessary to counteract catastrophic failures causing loss of service [Littlechild79]. The importance of the control program and the implications of its failure are clear; software reliability is paramount if the high standards of reliability and availability the telecommunications community has set itself are to be maintained; for example, an allowance is made for a maximum of two hours outage every forty years! [Scherer88].

A variety of 'switch based services' soon followed to take advantage of facilities made easily accessible by software control. These early services mirrored previous hardware based switch services and were normally just functional extensions to enable service providers to extract more value from existing services like POTS (plain old telephone service); Call Waiting and Call Answering are examples of such services. These facilities were, and still are, the basis of 'switch based services'. In this scenario manufacturers were in control of the development and delivery of new services [YoungJ88] , and could virtually hold operators to ransom for new service development costs because operators were locked into using what were indispensable propriety systems [Shapiro99].

3.3 Telecommunications software architectures

This section describes several standards based telecommunications software architectures in approximate order of their creation. It evaluates the possibility of each defining a generic and re-usable means of accounting for newly created services. Subsection 3.3.1 discusses IN, outlining its history and objectives before describing its architecture and evaluating it. It is shown that while the architecture was moderately successful in facilitating the creation of new services it was network technology specific, limited by the state of the art in software engineering and failed to take operations issues sufficiently into account. Subsection 3.3.2 discusses an operations architecture, TMN, the 'Telecommunications Management Network' standards for telecommunications management systems. TMN's origins and history are outlined, along with its objectives and architecture. In its evaluation it is shown that while the architecture was moderately successful in supporting standardised and well defined services, it lacked the concept of service creation and failed to encourage standardised software re-use, thereby denying management support for new services. It was also limited by information models and protocols which had become largely telecommunications specific. The last subsection discusses TINA, the 'Telecommunications Information Networking Architecture', an architecture which combined management and service issues, evaluates its progress to date and explains its use as the architecture underlying this research.

3.3.1 The ‘Intelligent Network’

The ‘Intelligent Network’ (IN) was first described in 1985 [Hass88]; it was the first significant attempt by the telecommunications industry to specify a service platform and architecture. While its services can be seen as quite basic in software terms, IN was the first standards based approach to facilitate software based service deployment on public telecommunications networks. The IN architecture still underlies many current telecommunications service implementations, e.g. GSM location registers [TINAB99].

3.3.1.1 History

As competition was introduced into telecommunications, first in the United States and later in other markets, service providers found it necessary to speedily deploy new services to differentiate themselves from their competitors and to respond to customer requirements. Service providers found themselves increasingly at the mercy of equipment manufacturers (‘switch vendors’) who remained in total control of the development and deployment of new services [YoungJ88]. Often there was a clash of interests: the equipment manufacturer could be offering subscriber products which were substitutes for the very services that service providers wanted them to implement! An example of this is the ‘Centrex’ service, where local service providers offered a bundle of services providing a ‘virtual’ PBX (private branch exchange) and equipment manufacturers offered actual PBXs installed on a subscriber’s premises.

In response to what could charitably be regarded as tardiness on the part of their equipment manufacturers, several regional Bell operating companies (companies providing local telecommunications services to regions of the United States, i.e. service providers) joined with Bellcore, their central research and standard setting body, to produce a standards based service architecture; the ‘Intelligent Network’. Standards were seen by operators as a means to extract themselves from their ‘lock-in’ to specific switching platforms by promoting competition between manufacturers and facilitating manufacturer independent service creation and deployment.

The term ‘Intelligent Network’ is a blanket term for a number of telecommunications industry and standards body initiatives that started in the mid 1980’s. The first intelligent network proposals owed their existence to an *ad hoc* industry consortium [Hass88]; this produced an architecture subsequently known as IN/1 in 1985. IN/1’s intent was to facilitate rapid service introduction by separating service logic from switches, however the major software elements were service specific and needed to be developed anew for each new service [Gansert88]. IN/2 followed soon after in late 1986.

In its time, the ‘second generation’ intelligent network was billed as a near panacea [Humes88]; it was specifically designed to enable the rapid development and deployment of new services [Gansert88], [Berman92]. It augmented the IN/1 architecture by expanding the set of switch and service capabilities, defining ‘Functional Components’ (FCs) to facilitate rapid service creation and adding better user interface capabilities through an ‘Intelligent Peripheral’ connected to the switch. It was soon realised that IN/2 was “an overly ambitious proposal which would entail unacceptably high risks and could not be implemented in a sufficiently short time” [Berman92]. The risks were mainly financial, not technical.

Based on this realisation, Bellcore released the IN/1+ architecture; this incorporated the ‘Functional Components’ and ‘Intelligent Peripheral’ of IN/2, but used just a subset of the functions specified - those needed for voiceband services. IN/1+ was to:

- Be attainable by 1991.
- Offer a profitable set of service opportunities.
- Offer an evolutionary step towards IN/2 [Gansert88].

As a short term solution, IN/1+ soon revealed its limitations; concern grew about the load on the signalling network that was also used to support the new services’ switch independence [Pierce88].

In the late 1980’s the IN/1+ initiative was shelved, and an industry forum called the Multi Vendor Interaction (MVI) was convened [Berman92]. This was launched in early 1989 and produced standards in 1990; the initial standards called Advanced Intelligent Network (AIN) Release 1, unified the many IN/1 architectures (relabelled ‘Release 0’ architectures) which by then existed. ‘AIN Release 1’ was an aspirational standard; it was seen as a model towards which the IN would evolve - its architecture was highly correlated with IN/2. ‘Release 0.1’ and ‘Release 0.2’ were introduced as realisable intermediate architectures towards ‘Release 1’; 0.1 introduced a common formal call model and 0.2 ISDN (integrated services digital network) announcement capabilities and default routing amongst other features. AIN Release 2 was announced in 1995.

Concurrent with these standards set in the United States' telecommunications market, the CCITT (*Comité Consultatif Internationale Télégraphique et Téléphonique*, now the ITU-T International Telecommunications Union – Telecommunications Services Sector) in conjunction with ETSI (the European Telecommunication Standards Institute), attempted to standardise the IN for other service providers [Garrahan93]. It defined Capability Sets (CS), with CS-1 roughly equivalent to AIN Release 1, and constructed a coherent IN conceptual model [Duran92]. Bellcore has since migrated its standards to use the terms defined in the ITU-T standards.

3.3.1.2 Objectives

This section outlines the overall objectives of the IN initiatives.

- **A market for telecommunications equipment**

By specifying a minimum functionality that they expected telecommunications equipment to have, telecommunications service providers hoped that price based competition between equipment vendors would help drive their costs down.

- **Rapid service introduction**

The key objective of IN/1 was "the ability to rapidly and flexibly add new services without requiring upgrades to the embedded switching system software" [Homayoon88]. IN/1 facilitated rapid service introduction by allowing centralised service deployment into a telecommunications network [Gilmour88], covering a "wide geographic area" (namely a U.S. regional operator or other administrative domain); it also allowed services to use the facilities of a number of switches.

It was thought that the 'Service Control Point' (SCP), the primary architectural element enabling this flexibility, would provide a suitable basis for the introduction and evolution of new services, but that ultimately, when proven (both technically and economically) the services would be incorporated into the switch [Head88] due, amongst other things, to the performance overhead added by using a separate service control point.

- **Service independent capabilities**

IN/2 sought to "reduce the interval between new service introduction", i.e. to reduce the time needed to create new services. Its key means to realise this objective was the specification of service independent capabilities that could be re-used in the definition of new services. A subset of these capabilities was adopted for IN/1+ [Bauer88].

- **Easier service creation**

Ultimately, it was recognised that it was better to support service creation at service provider rather than manufacturer level [Pinkham88], [YoungJ88], [Bauer88], [Berman92] and that “the success of IN [depended] on the ability of operator company personnel to create new services” [Pierce88].

3.3.1.3 Architecture

The key elements of the IN architecture are outlined in Figure 3.1. This figure uses the symbols commonly used to describe the IN architecture as used in [Gilmour88], [Weisser88] and [Berman92], and combines the elements described therein.

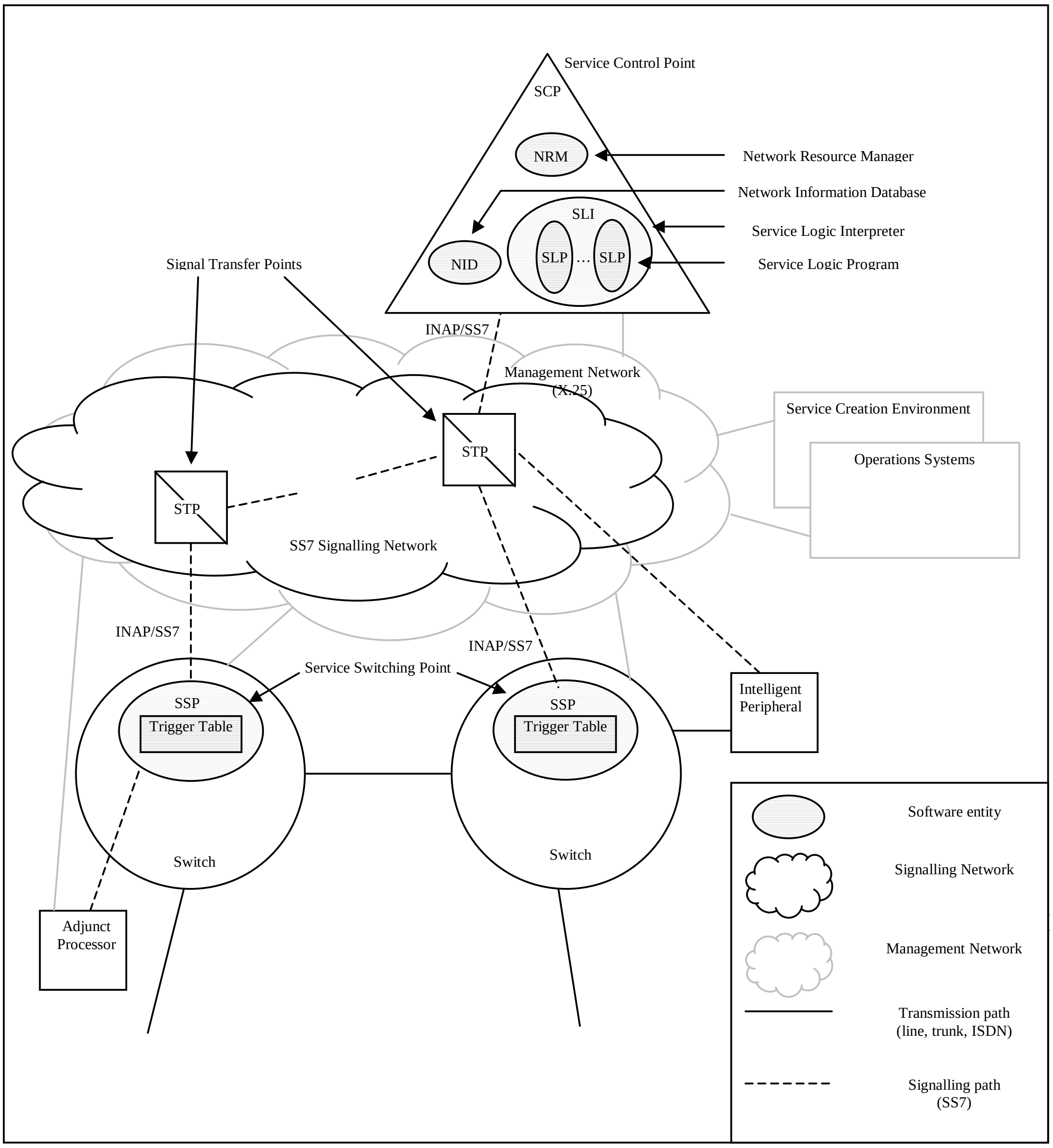


Figure 3.1, The IN architecture

A brief explanation which outlines the function of each element in Figure 3.1 follows. It has been noted that the relative simplicity of the IN is obfuscated by the abundance of acronyms which appear to serve to confuse the “uninitiated”! [Mercury93]. In deference to this, the explanations offer approximately equivalent, and more common, terms for many of the elements described.

The invocation of an IN service starts by the detection of a ‘trigger’, or event, at predefined ‘trigger detection points’ within the call (the specification of these detection points thus required the definition of a generic IN call model). When a trigger is detected, normal call processing is suspended, and information in the trigger table is used to formulate a query to the ‘Service Control Point’ (SCP). The query is expressed in a special purpose protocol (the IN Application Protocol (INAP)) then sent to the SCP over the signalling network, through ‘Signal Transfer Points’ (STPs), or packet switches.

The ‘Service Control Point’ processes this query and can either return a set of instructions to the ‘Service Switching Point’ in the switch, or execute them in its own ‘Service Logic Interpreter’ (SLI), with performance and signalling network traffic implications influencing the choice of execution environment [Gilmour88]. The set of instructions is called the ‘Service Logic Program’ (SLP). (The ‘Service Logic Interpreter’ was defined in IN/2 - prior to his IN services did not share control facilities; each service had its own ‘Service Control Point’ [Gansert88]).

The instructions consist of a set of ‘Functional Components’ (FCs) to be executed for the call, these are defined as “elemental network call processing actions” [Gilmour88]. (Example instructions include those to control call ‘legs’ (e.g. *create*, *join, free*) and those to give and receive information from call participants (e.g. *send* and *receive*) [Bauer88]). The ‘Functional Components’ were the key to service independence; as primitives they were designed to be reusable at quite a high level and well defined enough to be externally invokable on a ‘Service Switching Point’ (SSP) [Bauer88].

The other elements in the architecture include the ‘Adjunct Processor’: this is similar to the ‘Service Control Point’ except that it is not accessible through the signalling network, and is therefore not shareable: it has a direct communication link to the switch, and the ‘Intelligent Peripheral’, which offered an “enhanced” user interface to IN services through voice synthesis, announcements, speech recognition and digit collection [Berman92]. Logical elements included the ‘Network Information Database’ (NID) which kept information about access lines and trunks, and the ‘Network Resource Manager’ (NRM) which provided a location function enabling the correct switch to be invoked to continue call processing [Gilmour88].

The ‘Service Creation Environment’ provides an environment to compose IN services from ‘Functional Components’, it is usually present in an IN administrative domain, but not subject to IN standardisation - it is normally supplied by the SCP vendor, i.e. it is proprietary [MaA94a], [Kockelmans95].

‘Operations Systems’ offer support and management services variously called Operations, Administration and Maintenance (OA&M) [Bauer88], Operations, Administration, Maintenance and Provisioning (OAM&P) [Glitho95] or simply Operations Support Systems (OSS) or Operations Systems (OS). Some of these management systems are described in section 3.3.2

3.3.1.4 Evaluation

This section outlines the achievements and deficiencies of the ‘Intelligent Network’ architecture as defined and deployed in telecommunications networks throughout the world.

IN is technology specific. Most IN telecommunications services can be regarded as supplements to basic telephony service [Hellemans96]. In common with other telephony services, IN services must suffer the limited nature of their user interface [MaA94a]. Service processing is dependent on the detection, by the ‘Service Switching Point’ within the switch, of ‘triggers’ at ‘trigger detection points’ in the context of a ‘call’ prior to connection set-up, i.e. services are invoked for the end user by the transport provider [LaPorta97]. Within the switch, the call is modelled by either of two finite state machines: the Originating and Terminating Basic Call Models (BCM) [Berman92], which are, by their nature, strongly telephony oriented - states include those for routing, set-up and release. Each state is called a ‘point in call’ (PIC) and has an associated detection point where call processing can be handed over to a ‘Service Control Point’ (SCP).

The ‘Service Control Point’ has a standardised, generic, ‘Connection View’ of the call processing resources an IN switch offers [Berman92]. This standard model, while enabling some switch vendor independence, offers little in the way of transport technology independence to IN services; this was evidenced by the difficulty faced in modelling multiparty calls for IN Capability Set 2 (CS-2) - to enable such IN services as call waiting, call transfer and conference calls - because they necessitated changes to the connectivity of an existing call [O’RR98].

The physical separation of the ‘Service Switching Point’ (SSP) and the ‘Service Control Point’ (SCP), attempted to provide the reliability required for telecommunications services by provisioning expensive centralised facilities. IN did not attempt to distribute the service itself [Barr93] either by endeavouring to build a reliable system by redundantly deploying relatively cheap and unreliable facilities, or by any other means. This was reasonable given the state of the art in distributed systems at the outset of IN standardisation [Head88], but ultimately it leaves the IN looking like a legacy centralised system, by its nature more prone to catastrophic failure than a counterpart with distributed intelligence.

While failing to achieve a complete logical separation between a service and its underlying communications technology realisation, IN nevertheless established a physical separation between a service and its delivery [Hellemans96] that provided a useful basis for further service modelling work.

IN succeeded in creating a market for telecommunications equipment through standards, but detractors consider the inflexibility caused by over-specification in these standards acts to stifle innovation in offering new services [Isenberg98].

IN partially succeeded in its attempt to enable service providers to define their own services; switch vendors now sell ‘Service Creation Environments’ that take advantage of the ‘service independent’ facilities defined by IN. While it is unknown whether the overall costs of services developed using these environments is significantly less than the cost of purchasing services previously developed by switch vendors, service providers have been empowered to create and deploy their own services, albeit within the confines of proprietary environments [MaA94a], the currently deployed version of the IN Application Protocol (INAP) [TINAB99], and other technology constraints.

One of the reasons for the limited success of ‘Service Creation Environments’ was the advent of ‘feature interactions’ between IN services. In this context a ‘feature’ was either a service constituent [Zave93] or a simple service itself, used as a synonym because the term ‘service’ had become overloaded [Cameron93a]; similarly, the term ‘service interaction’ referred to the interaction of IN and switch based services (i.e. pre-IN services wholly implemented within the switch).

New services introduced into the public telecommunications network showed unwanted and adverse interactions [Griffeth93], [Zave93], [MaA94a], i.e. where the use of one service was altered by the use of another; for example call forwarding and call waiting [Cameron93b]. The problem was variously seen as one of incomplete system specification [Zave93], [Cameron93a] [MaA94a], of software re-use and maintenance [Griffeth93], [Cameron93a], of distributed systems (timing and race-conditions) and of artificial intelligence (dynamic resolution of conflicting end-user needs) [Cameron93a]. IN still exists, but the need to solve the problem of adverse interactions between service features has led to the adoption of contemporary computer science techniques, in a combined application of the approaches mentioned above, towards the definition of a wholly software based service architecture [TINAB99].

‘Service Creation Environments’ are also limited by the effects that the underlying service transport mechanism has on the flexibility of service definition. IN services are designed to use a dedicated underlying transport protocol (i.e. SS7), with the asynchronous IN Application Protocol (INAP) interactions facilitating the soft real time requirements of the signalling network. Unfortunately this requirement, while providing reliability, restricts the flexibility of service definition [Hellemans96].

IN neglected operations aspects of service provision, failing to specify standardised functional components that could be used to manage IN services. While the importance of management aspects was acknowledged [Bauer88], the IN initiative concentrated standardisation effort on service switching and control [Kockelmans95]. The creation of a software dichotomy ultimately proved harmful; later telecommunications service architectures recognised the need to support the concurrent development of services and their management facilities [TOCP95]. Separate management support for IN services is considered in the next section within the context of the telecommunications management standardisation effort.

3.3.2 The ‘Telecommunications Management Network’

The telecommunications management network (TMN) was originally conceived as a separate physical network used to manage a telecommunications network (TCN) [Scherer88], later relaxed to a logical separation [Shrewsbury95] due to the costs associated with maintaining a distinct physical network [Glitho95]. The TMN was defined to allow standards based management of telecommunications resources like transmission systems, switches and IN service elements (all called ‘Network Elements’, NE), and to facilitate the communication of management information between these resources and systems used to support their operation, administration, maintenance and provisioning (called ‘Operations Systems’, OS). TMN is considered as it endeavours to provide the management support necessary for telecommunications services.

3.3.2.1 Background and history

With the advent of competition, and the accompanying regulatory pressures for incumbents, service providers and equipment vendors both realised that it would be necessary to lower the cost of providing telecommunications services [Glitho95]. Providers increasingly sourced their equipment from competing vendors, the later introduction of IN standards facilitating price based product discrimination and helping to create a commodity market for telecommunications equipment.

Although it was anticipated that lack of management systems support had the potential to delay the introduction of new services [Bauer88], the IN initiative expended little standardisation effort on the definition of such support, concentrating instead on the service switching and control aspects [Kockelmans95]. They neglected management aspects at their peril: when a telecommunications network was composed of equipment from a few vendors, managing it was relatively straightforward, but as networks containing ever more diverse elements were deployed, service providers saw the need for integrated management systems rather than “a plethora of incompatible point solutions” [Scherer88].

The introduction of competition also meant that telecommunications service providers could no longer accept or afford “inefficient and work-intensive operations and management practices”. This lead to an increased need for the automation of operations and maintenance tasks [Glitho95] and a consequent requirement for remote management of Network Elements. Of course, competition also meant that a delay in the deployment of a new service due to lack of management systems support was no longer acceptable either.

These factors motivated an international standardisation effort that began in 1985, and produced its first standards in 1988 [Shrewsbury95], [Pras99]. The first CCITT standard defined to alleviate these problems, Recommendation M.20, proposed a network separate from the telecommunications network itself, dedicated to managing it, to be called the ‘Telecommunications Management Network’ (TMN) (The insistence on the physical separation of the TMN and the TCN it managed was later abandoned for pragmatic reasons [Glitho95]).

In the late 1980s and early 1990s, different study groups within the then CCITT, now ITU-T, concentrated on creating separate international standards for both the IN (Q. series Recommendations) and the TMN (M. series Recommendations). The IN standardisation effort towards ‘Capability Set 1’ (CS-1) chose to concentrate on the specification of service switching and control for ostensibly pragmatic reasons; to keep to the 1992 deadline for the production of the standards [Kockelmans95]. Recommendation M.3010, the TMN standard, appeared in 1992, with several significant additions to that produced in 1988, including the definition of an information architecture and a logical layered architecture [Pras99].

Unfortunately the division of standards between service control and management did not appear to be a natural one. While the IN initiative concentrated on fast service creation, the systems supporting service creation and management could only be based on proprietary solutions [MaA94a], [Kockelmans95]. It soon found that, as anticipated, the lack of standardised OS support was hampering service creation efforts. While IN attempted a "rapid" service development cycle of six months in 1992, service providers found that "OS enhancements traditionally [entailed] software generic development cycles comparable to that of pre-AIN switching systems"; once again service providers found themselves at the mercy of software providers, with "customer-funded software developments" of OS enhancements to support new services sometimes necessary [Pezzutti92].

Since then there have been efforts within regional standards bodies, specifically the European Telecommunication Standards Institute (ETSI), with a view to later international acceptance, to converge the standards to bring the elements of the IN within the purview of the TMN [Appledorn93], and to provide management support for new services defined outside the IN framework, including ISDN services. A further revision of the TMN standards appeared in 1996, and a further revision is due in 2000.

3.3.2.2 Objectives

This section outlines the overall objectives of the TMN standardisation effort.

- **A separate management network**

Telecommunications has traditionally used separate networks to support network management [Scherer88]. The telecommunications network itself and the signalling network were both deemed unsuitable because of the nature of management services and their network requirements. The definition of a standard basis for the interconnection of network elements and operations systems was the original motivation for the TMN standards; the TMN was primarily defined as a communications concept [ITU99].

- **Standardised management interfaces**

The TMN initiative attempted to standardise some of the functionality and many of the interfaces of the management network [Shrewsbury95]**.** As stated in [Appledorn93], "the primary purpose of the TMN is the definition or management interfaces", indeed the bulk of the TMN standards deal with interfaces [Glitho95].

- **Reduction of costs**

By enabling the automation of more network and service management functions, the TMN hoped to help reduce service provider costs. This was made evermore urgent by the advent of competition in many telecommunications markets [Glitho95].

- **Standardised information exchange between administrative domains**

As competition increased in telecommunications markets throughout the world, so did the number of service providers that needed to exchange management information. After the study period which culminated in the 1992 Recommendations, the TMN standardisation effort began to concern itself with the definition of the interfaces and information exchanged between administrative domains, i.e. the domains of different service or network providers [Prepare96]. Examples of the information exchanged over these interfaces include 'trouble management' (i.e. fault related), provisioning and restoration information [Shrewsbury95].

3.3.2.3 Architecture

The TMN describes several inter-related architectures; the functional, physical, and information architectures, and a logical layered architecture describing a hierarchy of management responsibility.

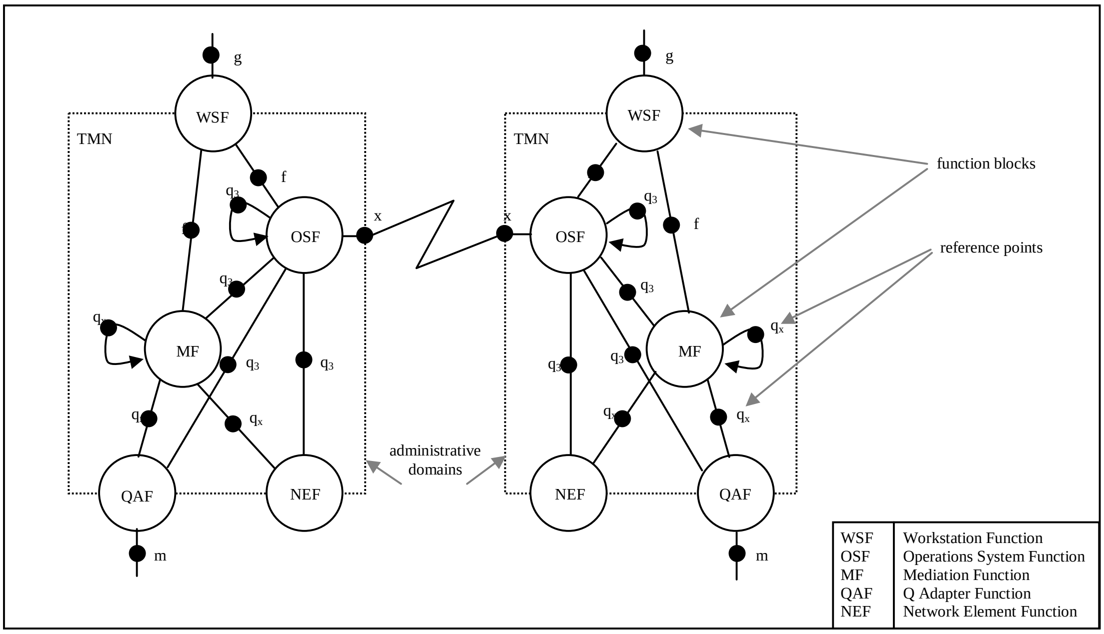


Figure 3.2, TMN functional architecture (based on Figure 5 in [M.3010]).

The ‘functional architecture’ specifies ‘function blocks’ which indicate the management functions that a TMN ought to support and ‘reference points’ that serve to delineate these functions. Together these form a reference model for the TMN which can be used to create a reference configuration [Shrewsbury95]. Figure 3.2 shows a reference configuration illustrating the function blocks and the types of connections between these blocks (each type represented by a reference point). These function blocks are not necessarily all present in a TMN, although most will have an implementation of the ‘Operations System’ and the ‘Network Element’ function blocks. The ‘Network Element’, ‘Q Adapter’ and ‘Workstation’ function blocks are drawn to show that only part of their specification is subject to TMN standardisation.

The ‘Network Element Function’ represents the management view of telecommunications network elements like switches and transmission systems. The ‘Q Adapter Function’ allows the connection of non-standard network elements or operations systems to the TMN, while the ‘Workstation Function’ represents the presentation of information to an end user. These function blocks have corresponding reference points (m and g) that are only partially described as they are outside the TMN.

The ‘Operations System Function’ and the ‘Mediation Function’ fall completely within the scope of the TMN standards. The ‘Operations System Function’ represents any system that processes information related to management, while the ‘Mediation Function’ stores, adapts, thresholds and filters information being transferred between Network Elements and these systems [Shrewsbury95]. The ‘f’ reference point allows for the attachment of a WSF to an MF or OSF, while the ‘x’ reference point represents the interconnection of OSFs in different administrative domains.

The remaining reference points (q_3 and q_x) are explained in their instantiations as interfaces in the explanation of the physical architecture which follows.

The TMN ‘physical architecture’ is defined separately to make an allowance for nodes which can contain functionality corresponding to more than one function block (commonly NE and OS functionality) [Glitho95]. The nodes that make up the TMN physical architecture are named after the primary function that they implement; thus a ‘Mediation Device’ must implement a ‘Mediation Function’ at least. The interfaces of the physical architecture are a realisation of the reference points in the functional architecture; the relationship is made explicit in notation by a capitalisation of the corresponding reference point, thus the ‘Q_3’ interface realises the ‘q_3’ reference point, while the ‘Q_x’ interface realises the ‘q_x’ reference point.

The ‘Q_3’ interface is a definition of an OSI Common Management Information Service Element (CMISE), defining the link between a manager/agent* pair communicating using the Common Management Information Protocol (CMIP). The ‘Q_x’ interface is sometimes referred to as “the Q_3 interface’s underdeveloped brother” [Glitho95]; it is so named as originally two ‘q’ reference points, ‘q_1’ and ‘q_2’ were defined, their definitions later merged to ‘q_x’ [Pras99]. It was intended to be used when cost or efficiency considerations forced a subset of CMIS to be implemented, unfortunately there was little agreement as to what could be omitted [Glitho95]. A sample physical architecture is shown in Figure 3.3

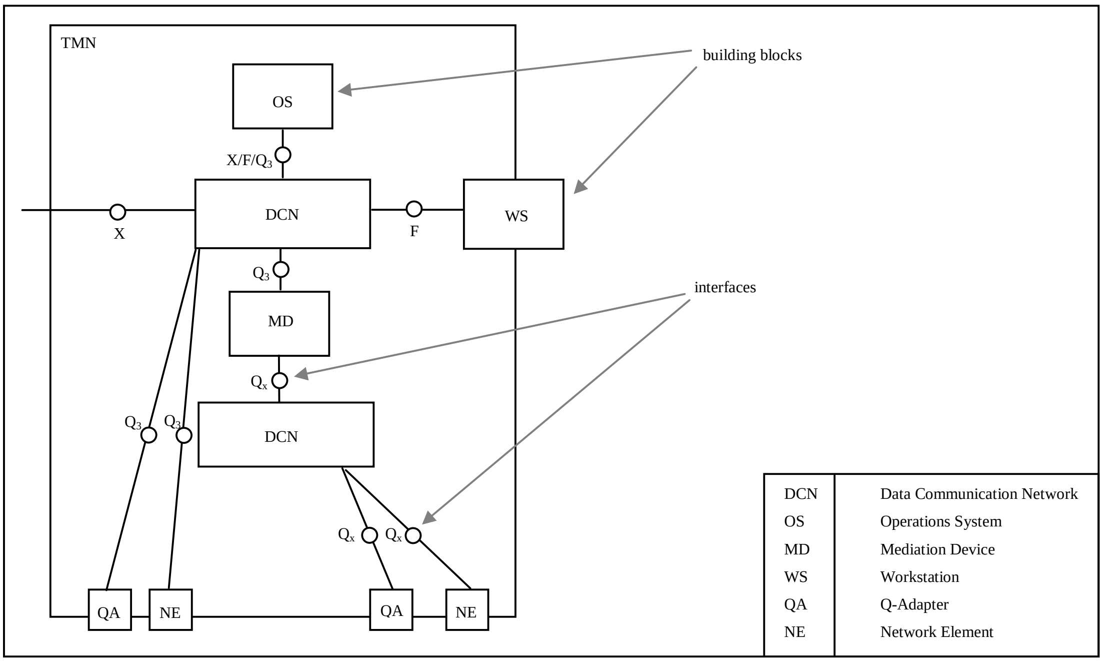


Figure 3.3, A sample TMN physical architecture (based on that in [M.3010])

At the lowest level, the TMN ‘information architecture’ is based on that of OSI systems management, itself a union of object orientation and the manager/agent paradigm. OSI systems management defines ‘managed objects’ that differ from conventional software objects in that their access methods do not reside entirely within the object, but in the agent, which provides a set of common operations known collectively as a service [Shrewsbury95].

At a higher level of abstraction, the TMN standards also define a ‘logical layered architecture’, adopted from BT’s ‘Open Network Architecture’ [Pras99], where it was used to structure OS functionality [Senior91]. This breaks management tasks into subsets, then divides them into layers in an attempt at data hiding; revealing the right level of detail and controlling the quantity of information passed to the layer above. This concept also facilitates

---

* An OSI systems management term for a relationship between entities approximating that between a client and a server.

inter-domain management as only the appropriate level of operational detail needs to be passed to other administrative domains, i.e. potential, if not actual, competing service providers. Figure 3.4 shows a typical hierarchical decomposition of management functionality. Note that while the ‘q type’ reference points identify possible physical interfaces [Appledorn93], this need not necessarily be the case; they could be implemented within the same OS entity.

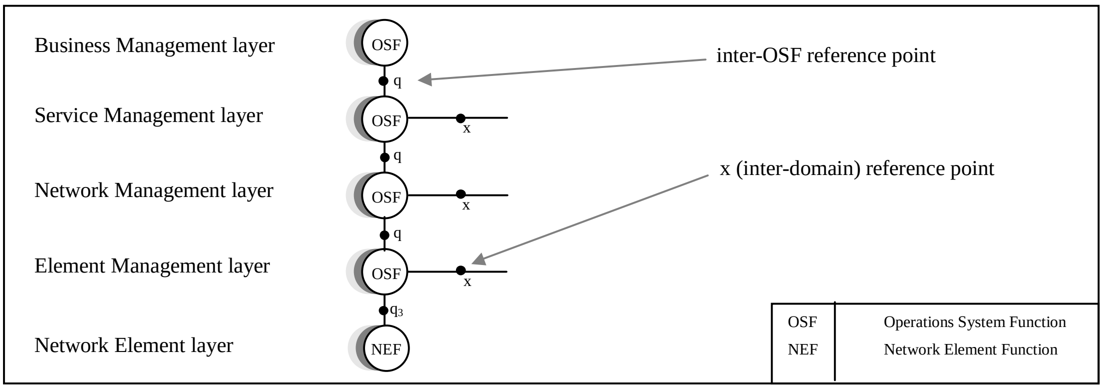


Figure 3.4, A typical layered functional hierarchy

3.3.2.4 Evaluation

This section evaluates the effect the TMN standardisation effort has had on the definition of software based telecommunications services.

TMN lacks the concepts of service creation [Appledorn93], [Kockelmans95] and application building blocks [Kockelmans95], both of which were used in contemporary IN service definition and were therein important software re-use concepts. This made it difficult to provide the management support needed for the rapid development of services [Kockelmans95]. TMN does describe ‘Management Service Components’, used to help specify management services as part of management applications [M.3200], but lacks a component architecture that would allow software built to these specifications to be re-used. While in the past a TMN specific component architecture might have been developed to address this, the authors of [Davison99] thought it “important to consider if this issue is sufficiently unique to telecommunications management to warrant a reliance on TMN standards for its solution”.

Service creation was also hindered by the TMN standardisation effort’s “reluctance to standardise the functionality of nodes” as it was thought this would “constrain product offerings”, instead they concentrated on enabling the definition of standard interfaces which would allow nodes to inter-work “as long as the protocols [were] specified to a level that [allowed] applications to interact” [Glitho95].

The reluctance to standardise TMN functionality was partly due to the selection of the OSI systems management standards (X.700 series) as the basis for TMN interfaces [Glitho95]; OSI standards already provided specific management functions, i.e. the fault, configuration, accounting, performance and security 'system management functional areas' (commonly knows by the acronym 'FCAPS'). TMN mapped its telecommunications related functions to these more general systems management functions (with the exception of OSI security management, which had no TMN equivalent at the time of the original mapping [Scherer88]). These 'systems management functions' were dependent on the OSI management service and protocol, CMISE and CMIP, which meant that the protocol dependence was imported into the TMN. (Although it was noted that "TMN [was] not permanently tied to X.700 OSI management methods" [Shrewsbury95], the selection of CMIP to realise the q_3 reference point resulted in a resolute association between the standards; in effect, TMN was "tightly coupled with OSI management" [Glitho95]).

Recognising the lack of functional specifications, several bodies subsequently produced specifications, including the then 'Network Management Forum' (NMF, now known as the 'TeleManagement Forum (TMF)), which provided specifications for testing, trouble and security management amongst others [Shrewsbury95] and the 'Netman' project, part of the 'Research and Development in Advanced Communications in Europe' (RACE) initiative, which produced 'Common Functional Specifications' for a number of areas, including accounting management [Smith92], [Netman94]. The lack of TMN functional standards has made it necessary for several TMN platform vendors to supply the functionality in proprietary ways [Davison99]. Recent TMN efforts have seen a reversion to the specification of a protocol independent framework for telecommunications management [TMN00].

Lack of architectural protocol independence [Davison99] also caused problems when attempting to integrate 'legacy' management systems; while Q-Adapters were specified for this purpose, in reality they were "difficult to develop" [Glitho95] and the architectural prescription of the "protocols and information specification techniques" was seen as a "problematic constraint on the applicability of TMN" as early as 1996 [Prepare96]. Knowledge of TMN technologies is not "IT mainstream" [Davison99]; given this, and the lack of tool support, many service providers found it "difficult to justify expense of migrating to new interfaces" [Glitho95]. As a consequence, today fewer than half of the respondents to a commercial survey quoted in [Davison99] had fully or partial compliant OSs.

When it came to defining interfaces there was [Glitho95], and is [Davison99], a lack of adequate tool support, with the OSI systems management based information model chosen seen by some as excessively complex [Davison99], its requirements found conflicting and difficult to meet [Glitho95].

The ‘logical layer’ concept adopted from BT’s ‘Open Network Architecture’ by the TMN [Pras99] was found to be a useful principle “for separation of management scope and responsibility” [Prepare96] while it is considered “the most important concept of TMN” by the authors of [Pras99]. Recent TMN standardisation work has tended to concentrate on the service and business management layers, with a recognised need for standards to automate “the activities that occur between administrations” [Glitho95], although in [Davison99] the authors question the use of “non-generic approaches to supporting the automation of business transactions between companies” given the growth of electronic commerce.

3.3.3 The ‘Telecommunications Information Networking Architecture’

The ‘Telecommunications Information Networking Architecture’ (TINA) was a combined effort by service providers and their equipment and software suppliers to produce a software based service architecture embracing the concepts of distributed computing and object orientation. It also hoped to provide migration paths from established technologies like IN and TMN and to provide a greater integration of the facilities previously provided by these technologies.

3.3.3.1 Background and history

TINA was propounded by a consortium consisting of service providers and their suppliers of software, computer and telecommunications equipment. This alliance was formally announced in an international symposium in Japan in late 1992 [Dupuy95], and began its work in early 1993 [Pavon96]. The formation of the consortium reflected some exasperation and impatience with telecommunications standards setting procedures, and bodies, felt to be out of touch with the new market realities [Marshall95]; reflected in particular in the four year time frames such bodies typically allotted themselves for the specification of standards.

The consortium proposed to produce *de facto* standards: specifications for the design and construction of telecommunications applications and a set of concepts and principles to guide such design [Pavon96]. These specifications, principles and guidelines constituted the ‘TINA architecture’: amongst other aims, it hoped to integrate IN and TMN concepts, recognising that is was no longer feasible “to support two independent architectures while applications on both architectures must inter-operate” [Appledorn93].

The architecture was informed by advances made in distributed computing and object orientation within and without prior research projects and prototype work carried out separately by consortium members. It was envisaged that, rather than having many small groups working on similar issues, co-operation would enable “more effective and rapid progress” [Dupuy95]. With this in mind, a ‘core team’ of about forty engineers was assembled at Bellcore to specify, validate and refine the architecture with the aid of sample services. The first two architectural drafts were delivered to member companies at the end of 1993 and 1994, others following in March 1995, October 1996 and January 1997. The last two versions represented a stable architecture, with minor restructuring necessitating the final release [TSA97]. Their work was subsequently made available to the wider research community.

The TINA architecture adopted and integrated concepts from several sources, including: ITU standardisation efforts including IN, TMN and ODP standards [Dupuy95]; research projects including Bellcore’s ‘Information Networking Architecture’ (INA) and the related ‘Touring Machine’ demonstrator [Nat92], [Gopal92], [Arango93], [Rubin94], the RACE (Research and Development of Advanced Communications in Europe) project ROSA (Race Open Services Architecture) [Mierop93], [Hall94]; and commercial architectures like the Bellcore (now Telecordia) OSCA (Operations Systems Computing Architecture) [Nat92], ANSAware [Herbert94], [Adler95] and OMG’s OMA [Kitson95].

3.3.3.2 Objectives

The TINA Consortium sought to define a consistent “software-based architecture for future information networks” [Barr93] that would support “open” telecommunications and be capable of validation [Dupuy95]. The objectives[*] for this architecture, the ‘TINA Architecture’, included:

- **Support for new services**

The TINA architecture aspired to support the rapid production and deployment of new services [Rubin94], [Dupuy95], [Pavon96]. The most important requirements for such support included those for software interoperability and re-usability. The consortium recognised that “if the interoperability of application and platform software is not provided by the adoption of

---

[*] TINA shared many objectives with the Bellcore INA initiative, indeed the INA initiative provided many of the early architectural specifications to the TINA Consortium [Rubin94]. For this reason some of the explanations of the objectives listed cite references to sources describing the INA as well as those describing TINA.

the architecture, interaction between products of different suppliers would be achieved by *ad hoc* solutions with additional costs, possible delay in deployment and lack of reliability." [TReq95].

To support re-usability, the architecture was to specify component based software resources that could be used to construct services. It was hoped that some of these could be made available for new services in a library or toolkit [Dupuy95]. (These components were similar to INA 'Building blocks' [Rubin94], themselves related to the 'cluster' concept of ODP and ANSAware [Herbert94]). Run-time re-use of application software was envisaged: "The architecture should enable run-time re-use of application software so that the software can be accessed in providing new services and management capabilities" [TReq95]. A TINA component was to consist of several 'Computational Objects', or servers, each capable of supporting several interfaces. An interface that supported functionality visible outside a component was called a contract.

TINA also attempted to specify a common framework and methodology to simplify service design. Part of the support for new service types was a requirement for the "concurrent creation of service and management facilities" [TReq95].

- **Support for service distribution**

By not prescribing the location of software components, the TINA architecture attempted to enable the flexible placement of platform and application software. The architecture also strove to facilitate transparent distribution of application software on nodes that supported such transparencies [TReq95].

- **Telecommunications network technology independence**

A major objective of TINA was to provide service designers and implementers with usable abstractions of the telecommunications network that could be used to design services without building in technological dependencies. This was stated as a requirement for the "de-coupling of applications from computing and network hardware resources." [TReq95].

- **Co-ordinated management**

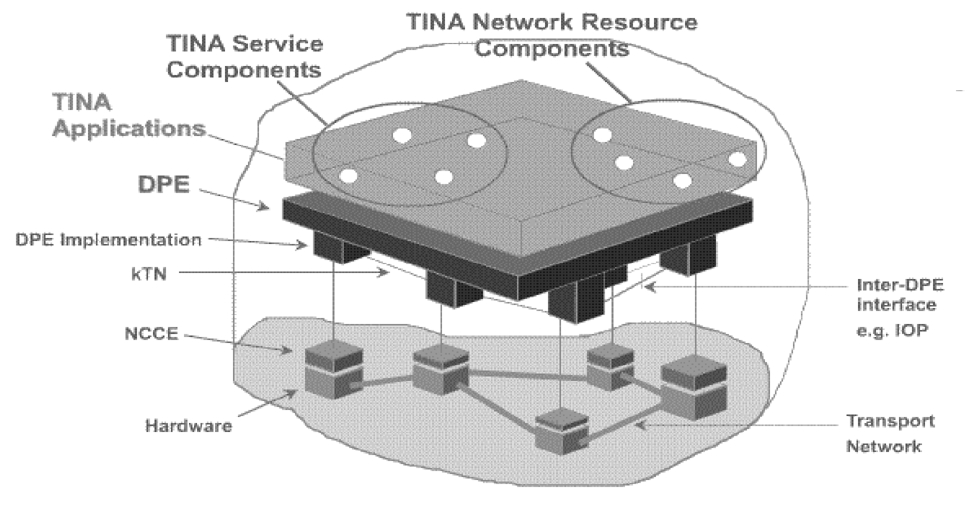
The TINA architecture was to enable the co-ordinated management of software components and network resources, allowing for the probability of multiple vendors for both [TReq95]. It was hoped to integrate the concepts of both the IN and TMN initiatives [Appledorn93], [Pavon96]. The TINA architecture also endeavoured to support an ‘evolutionary’ migration from these architectures.

3.3.3.3 Architecture

The TINA architecture consists of several sub-architectures that have complex interdependencies. TINA defines the ‘Overall TINA architecture’ and several constituent sub-architectures, namely: the ‘Service Architecture’; the ‘Network Architecture’; the ‘Computing Architecture’ and the ‘Management Architecture’ [TOCP95]. The architectures are specified independently, but in reality they depend on concepts defined in each other, which makes their independent description difficult. The ‘Service Architecture’ is the most relevant to this report, but the pertinent concepts from other sub-architectures are also defined.

The ‘Overall TINA Architecture’ defines concepts and principles it believes should be used to structure telecommunications software. The ‘architectural layers’ and ‘architectural separations’ it defines are deemed applicable to all the sub-architectures. TINA identified two orthogonal layering principles: computing and management layering.

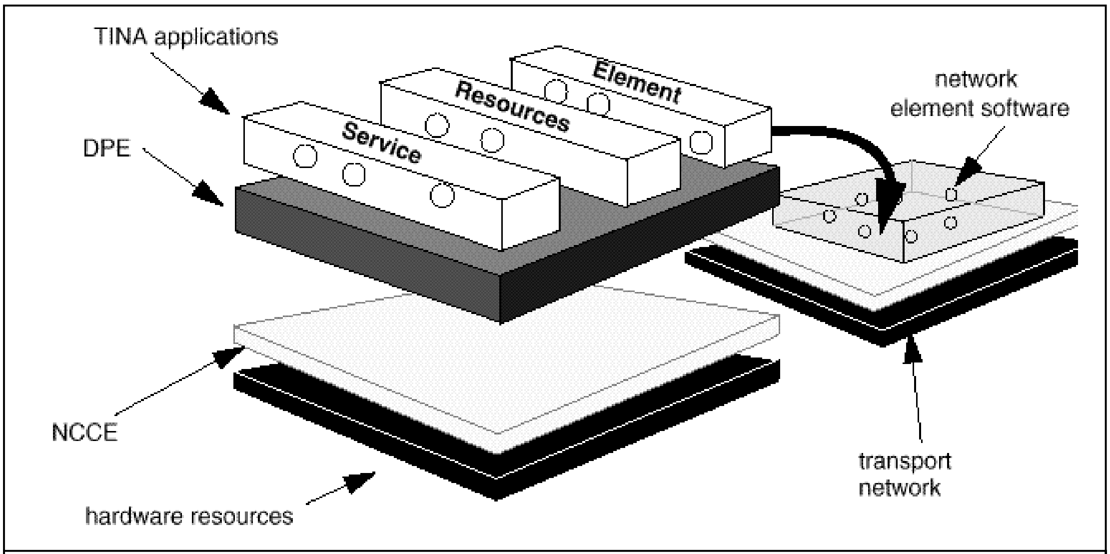
‘Computing layering’ refers to the separation of telecommunications applications and the underlying distributed processing environment (DPE), with its supporting layers, as shown in Figure 3.5. Note that the transport network shown in this figure introduces a telecommunications specific layer. The overall architecture does not specify that all the network elements that comprise the transport network have DPE functionality, or that all DPE nodes have underlying transport network elements.



kTN	Kernel Transport Network	DPE	Distributed Processing Environment
NCCE	Native Computer and Communications Environment	IOP	OMG Inter-ORB Protocol

Figure 3.5, TINA computational separations (adapted from [Tt99])

‘Management layering’ is an adaptation of TMN layering (shown in section 3.3.2.3). Instead of the five management layers, TINA defines three: the service, resource and element layers. TMN layering was modified for two reasons [TOCP95]: TMN was primarily intended to be used in the management of transport networks, whereas other elements needed to be managed in TINA systems, and the layering, thus expressed, was also deemed suitable for software other than management software, for example services using the transport network. Management layering is shown in Figure 3.6



DPE	Distributed Processing Environment
NCCE	Native Computer and Communications Environment

Figure 3.6, TINA management layers (from [TSA97])

The overall architecture also defined “a composition model with a common sense separation of concerns” [Dupuy95] that could be used to design services and components [Berndt94]. This model was not made part of the service architecture because, while it was to be principally applied to service design, it was also deemed useful for components in the resource and element layers. It was also unproven [TOCP95]. The ‘Universal Service Component Model’ (USCM) is shown in Figure 3.7

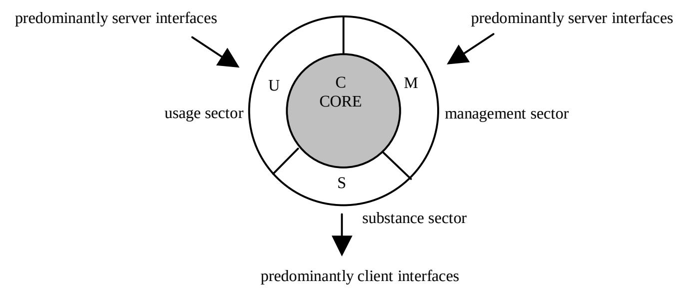


Figure 3.7, The ’Universal Service Component Model’

The ‘Core’ defines a service’s “primary value to a user” [Berndt94], i.e. the nature of the service or component functionality. The ‘Usage’ segment represents a service or component’s appearance to the end-user or another component, respectively. The ‘Management’ sector represents management functionality, while the ‘Substance’ sector represents the dependence upon external resources. For a service, these segments represent its ‘access layer’. The segments bear a resemblance to the layering principles defined in Bellcore’s OSCA (Operations Systems Computing Architecture), used by the INA initiative, except that there the layers, ‘User’, ‘Processing’ and ‘Data’, which roughly correspond with Usage, Core and Substance, were only used to partition building blocks, i.e. components, according to their function [Nat92].

The ‘Computing Architecture’ defines modelling concepts that are used to define software in TINA systems and the TINA DPE. It refines the ODP reference model to make it “suitable for the design of telecommunications systems” [TOCP95]. The computing architecture defines modelling concepts for the ODP Enterprise, Information, Computational and Engineering viewpoints. TINA adds the concept of the ‘Stream Interface’ to the ODP concepts, where a stream interface is one without operations which facilitates the flow of structured information. TINA uses a superset of the OMG’s IDL, called the ‘Object Definition Language’ (ODL), as a notation for computational specifications. ODL contains additions to deal with multiple interface objects and stream interfaces.

The TINA ‘Management Architecture’ extends TMN with ODP concepts, and is built on the assumption that management components are deployed on the TINA DPE [Pavon96]. Management functionality is thus independent of the protocol used to carry the management information. The TINA Management architecture dictates generic management principles that should be used for all telecommunications management software; the five OSI functional separations, the application of the computing architecture to model management systems and the application of the service architecture to management services.

The ‘Management Architecture’ defines two sub-categories of management; ‘Telecommunications management’ and ‘Computing management’. Telecommunications management applies to three categories of components corresponding to each management layer defined in the overall architecture; the service components (defined in the service architecture), resource components and elements. Service management is defined in the service architecture by applying the generic management principles. Network management in TINA is defined in the ‘Network Architecture’ and applies to both the network management and element management layers defined by TMN, and thus to both resource components and elements. Computing management covers computing platforms, the DPE and the actual service software deployment, installation and operation.

The ‘Network Architecture’ provides concepts that describe transport network, and the functionality it offers services, in a technology independent way. It provides a “high level view of connections to services” [TOCP95] and generic descriptions of network elements in the ‘Network Resource Information Model’ [Lengdell96]. At the lowest level, an element proxy acts as a technology gateway between TINA and management specific protocols like CMIP. At the highest level of abstraction, offered by the resource layer to the service layer, a connection graph represents connectivity between stream interfaces. The ‘Communication Session Manager’, defined in the resource layer, translates a logical connection graph, at the service level, into a physical connection graph, at the transport network level. The elements of a connection graph are shown in Figure 3.8

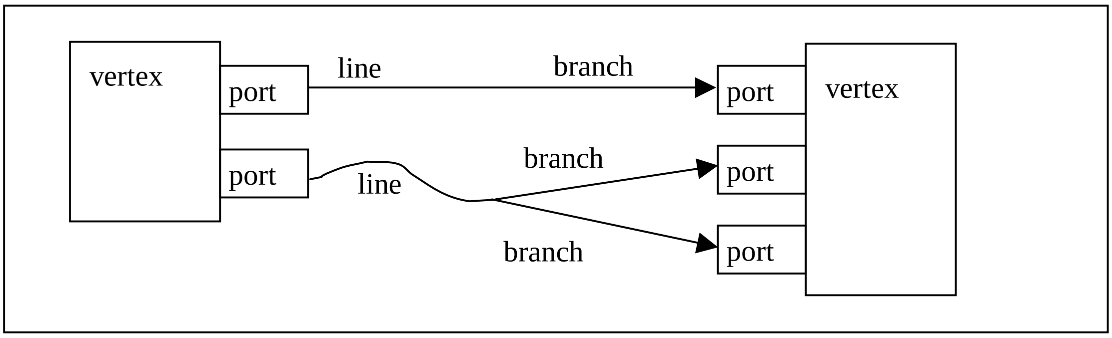


Figure 3.8, A TINA connection graph.

The TINA ‘Service Architecture’ [TSA95], [TSA97] attempts to define a set of concepts for the design, implementation, usage and operation of telecommunication services [TOCP95]. It identifies and defines generic service components that can be re-used to define new services [Dupuy95]. The service architecture extends and specialises the OSI management functional separations adopted by the management architecture, then applies them to service management. This means that management functionality is an integral part of the definition of an TINA service, not just management services. Figure 3.9 shows the TINA service architecture components and their interactions

The primary concept defined by the service architecture is the ‘session’, which is concerned with “temporal relationships and service activities” [TOCP95], i.e. it provides a context for user interactions with a service within a defined time period. The TINA service architecture identifies several session types, including; the access session, which represents a customised, mobile and secure access point to many services; the service session, representing a service activation; the usage (or user) service session, representing a single user’s interaction with this service activation; and the communications session, which maintains state (in the form of a connection graph) about the network connectivity of a service session.

The session concept was described in the Bellcore ‘Touring [sic] Machine’ project, where it was an abstraction representing “the control relationship among applications software providing services to, or acting on behalf of, users participating in a communications attempt” [Gopal92], although it can be seen that this archetypal concept conflates the now separate service, user and communications sessions. TINA defined several sessions because the separation promoted independence between different aspects of service usage. This independence meant that “the modification of a particular session model or related mechanism [did] not impact on models and mechanisms governing other types of session.” [Pavon96]. TINA sessions can also span several administrative domains; portions of sessions are defined that apply to each domain, for example the usage service session can be split into customer and retailer domain usage service sessions. The session concept offered several advantages over the ‘call’ concept; whereas a call was concerned with the allocation of communications resources, with service logic awkwardly invoked from predefined points within the call, the session could support service specific functionality ubiquitously [Pavon96]. The session concept was also originally thought useful in the detection of unwanted feature interactions [Arango93], [Pavon96].

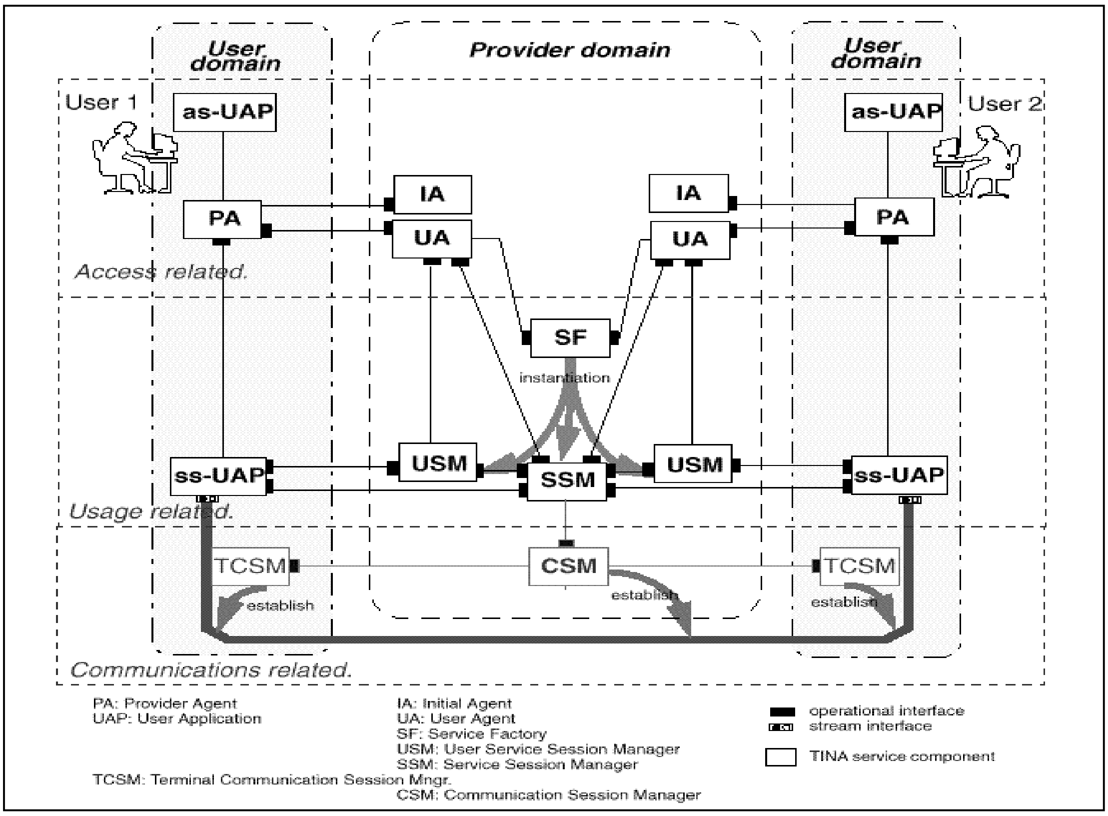


Figure 3.9, TINA Service Architecture components & interactions (from [TSA97])

The components that support the access session are the ‘User Agent’, and the ‘Provider Agent’, these allow the association of users and terminals with services and networks.

The ‘User Agent’ represents a service user in a provider domain and facilitates the creation or joining of service sessions. The ‘Provider Agent’ allows a user to access their user agent in a particular provider domain [TSA97]; this means that a particular service provider’s ‘Provider Agent’ must first be present in the user domain. The ‘Initial Agent’ was added to the set of service architecture components to support user mobility by allowing a user to contact a service provider from any terminal. The initial agent facilitates the creation of the access session which then exists between the provider and user agents, but does not participate further in this session. The ‘access session User APplication’ (asUAP) provides a user interface to provider agent functionality. Similar components to the user and provider agents were described in previous architectures using the names ‘User agent’ & ‘User’ [Mierop93], ‘User Agent’ & ‘Generic session end point’ (which combined both access and service functionality) [TSA95], and ‘User Agent’ and ‘Terminal Agent’ [TOCP95]).

The service session is supported by the service factory and the user and service session manager components. To create a service session, a user must first select which service to start using the access session components; this involves the provider agent requesting a list of subscribed services from the user agent and returning it to be presented by the access session user application.

When the service is chosen, the provider agent can start a service session user application (ssUAP). At this stage, a new service session can be requested of the provider agent by the service application, the provider agent relaying this request to the user’s user agent in the service provider domain. The user agent contacts the ‘Service Factory’ appropriate to the required service session and requests the creation of the ‘Service Session Manager’ and ‘User Session Manager’ components to support this session. The references to these components are then propagated back through the user agent and provider agent to the service session user application. At this stage a single user service session exists as the association between, at least, the service session manager, user session manager and the service session user application. The usage service session, representing a user’s view of a service is supported by the user session manager. Facilities exist to invite other users to join the service session and for the extension of already existent sessions to incorporate new users, mediated by the service session manager.

A service session usually requires the use of telecommunications network facilities. These facilities can be used within the context of a communication session, which represents a service view of bindings between stream interfaces supported by the service session user applications. Only one service session can be associated with a communications session at any one time. The ‘Communication Session Manager’ allows a service session manager or user session manager to set-up, modify and remove ‘stream flows’ as described by a logical connection graph, where a stream flow is an abstraction of a connection [TSA97]. The ‘Terminal Communication Session Manager’ allows the communication session manager to set-up, modify and remove connections in a user’s domain.

In general, the service architecture promotes a separation between service logic and resource provisioning, facilitating the provisioning of new services independently of the underlying network [Dupuy95].

3.3.3.4 Evaluation

Overall, the TINA Consortium as an entity cannot be judged as wholly successful, although the ideas propounded by it will probably continue to influence and inform many telecommunications service modelling and design activities. There are few, if any, commercially deployed telecommunications services that are entirely based on the TINA architecture; parts of it have been used, others parts adapted, still others ignored. Acceptance and use by a wide range of vendors is a necessary precondition for the success of any software architecture; in this sense the TINA architecture cannot be viewed as a success. Perhaps this was due in part to the consortium’s initial lack of openness: for those outside the consortium it often proved difficult to gain access to relevant documentation. Such was the case for the early part of this research.

At least part of the problem can be traced back to support for ‘legacy’ systems. This is incredibly important in any industry relying on software, but are especially important in service industries. A denial of service for upgrades or wholesale infrastructure replacement is unacceptable. Reliability and availability are at the heart of service provision; indeed some believe they provide one of the main differentiating factors between software applications and services [MaA94a]. Perhaps migration from legacy systems ought to have been considered more closely by the consortium, but inter-working with legacy systems is not commercially lucrative outside specialist integrators (there is only one specialised ORB vendor in the TINA Consortium), nor does it guarantee sales of new equipment.

When the TINA Consortium started, the idea of a ubiquitous distributed processing environment was probably fanciful. There were several architectures that specified DPEs. In hindsight the most influential of these was the OMG's Common Object Request Broker Architecture (CORBA), but in the timeframe of the early TINA Consortium even the adoption of CORBA did not guarantee inter-operability between applications written by different software suppliers. (Such inter-operability did not become a feature of CORBA until the release of CORBA 2.0 in 1995, with ORBs featuring such functionality appearing later again). It was gradually realised that telecommunications software was not such a special case; it too, could, and should, use 'mainstream' software engineering practices and products. With this in mind, the TINA Consortium has lately turned to directing telecommunications specific extensions to CORBA rather than designing and specifying yet another distributed processing environment.

Perhaps another reason for TINA's lack of success can be attributed to the sheer quantity, verbosity and complexity of the material that needs to be understood before forming a coherent view of the architecture: "Understanding all the documentation imposes as steep learning curve for parties interested in implementing TINA" [McKinney98]; this in an industry not known for its reticence!

It has also been commented that "the details needed to implement TINA do not exist and the current documentation leaves much room for interpretation" [McKinney98]. The selection of ODP viewpoints as a method to describe and specify TINA systems has also been criticised; it was felt that while they were a good documentation tool, they offered little construction guidance, e.g. in combining the viewpoints to create a system [TSDG95].

TINA has also been criticised for concentrating on connection oriented network environments [McKinney98]. The architecture's general applicability to connectionless services has been shown experimentally [Hwang96], [Lewis98], but some of the communication session related components in particular show a connection oriented bias; "TINA has still not been deployed in an actual Internet user setting" [McKinney98].

Experimental results have found that the architecture's support for software re-use as far from ideal; work carried out in the ACTS Prospect project [PD22B98], which based its sample services on the TINA service architecture, found that while some software and specification re-use was possible, true component based re-use was difficult. Ways to specify and package telecommunications software for re-use were subsequently explored in a follow on ACTS project, Flowthru [Lewis99].

3.4 Charging for telecommunications services

The requirements for both service fulfilment and pricing are demonstrated in the support a service architecture gives for service charging. This section considers the support for accounting and charging for service usage in each of the service architectures outlined in the previous section. It begins with an historical overview of telecommunications charging, explaining its provenance and its development into present day charging systems. It then describes the degrees of standards based support for telecommunications charging and offers some explanations for their present positions.

3.4.1 Background

As seen in the previous chapter, telecommunications systems, formerly predominantly telephony systems, have historically had high running and maintenance costs, associated with manual and mechanical switching systems respectively. Service providers used two methods to cover these costs in manually switched systems: metering and ticketing [Littlechild79]. With metering, a counter, or meter, associated with each subscriber was incremented for each call made, the increment executed by hand. With ticketing, the details of a call were written down on a ticket, later used as a basis for the calculation of a fee based on distance and duration. Metering was usually used for local calls and ticketing for long distance calls. The same methods were later automated for automatic switching systems.

When automatic switching was first introduced it provided only local connections: long distance services like subscriber trunk dialling (STD in Britain and Ireland) and direct distance dialling (DDD in North America) were later innovations. Some service providers, particularly those in Europe, employed a scheme called ‘message rate charging’, where each subscriber’s meter was automatically incremented for each call [Littlechild79]. When long distance dialling was introduced, modifications were made to provide ‘periodic pulse metering’ (PPM). With this system, the meter was incremented at intervals throughout the call, the duration of these intervals dictated by the call destination.

In North America, several service providers still used flat rate, unmetered, charging for local calls, where the line and equipment rental were set to cover the cost of service for an average subscriber. PPM could not be used where a subscriber’s line did not have meter, so equipment to provide automatic toll ticketing (ATT), later automatic message accounting (AMA), was introduced. In this scheme, the details normally noted by the operator were automatically recorded to be used as a basis for later charging. This scheme allowed service providers to produce itemised bills, but the associated processing overheads were borne by subscribers in the form of a minimum charge for each call as computing facilities were still relatively scarce and expensive.

The lack of transparency and consequent fraud potential of pulse metering eventually led to its replacement by many service providers. The adoption of ticket based schemes could provide the itemised bills that many subscribers requested, and could also justify the introduction of potentially lucrative minimum charges for local calls. Such systems produce 'Call Detail Records' (CDRs) which can trace their origin to hand-written 'tickets'.

3.4.2 Overview

This section introduces terms and concepts that can be used as a common vocabulary when describing telecommunications accounting systems. Each architecture tends to define its own terms to refer to common concepts, coining a vocabulary that can usually only be understood within its own context. The concepts defined within the OSI systems management standards are chosen because they formed the basis of several architectures. By its nature this section thus also outlines standardisation efforts in telecommunications accounting.

The OSI systems management 'Usage Metering Function for Accounting Purposes' defines a relatively intuitive three stage model describing "accounting for resource utilization" which consists of the following sub-processes [X.74298]:

- **Usage Metering**
  Usage metering defines the creation and persistence of 'usage metering records' (UMRs) generated as a consequence of the occurrence of 'accountable events'[*]. Several accountable events may result in the creation of a single usage metering record and "the use of several resources will generally give rise to several usage metering records".
- **Charging**
  Charging relates to the creation of 'service transaction records' (STRs) by combining usage metering records. The term service transaction is "used in its usual English meaning to denote things like a telephone call or the sending of a electronic mail message" [X.74298]. Pricing information based on resource usage is also usually added at this stage.

---

[*] An event is deemed accountable based on its presence as a 'reporting trigger' in any accountable object's control object (this is described in more detail in section 3.4.4).

- **Billing**

  Billing refers to the relation of service transaction records to a specific subscriber and the eventual production of a bill.

When regarded in a wider telecommunications industry context, billing usually also includes access charges (e.g. line rental), discounts and other periodic charges not related to resource usage. Some telecommunications industry forums, especially those with higher level service and business process oriented views, also use the term ‘billing’ to refer to the complete accounting process [TMFBoM98]. In this report ‘billing’ refers to the sub-process outlined above. X.742 specifically restricts itself to specifying “the activities and management information required to support the usage metering process”.

The TMN standardisation effort adopted OSI systems management recommendations as a basis for the definition of many management functions. Several of these general systems management functions, described in Recommendations X.730-X.750, were subsequently adapted and specialised for the management of telecommunications networks. One such specialisation was Q.825 which described three new functions to support ‘Call Detail Recording’ for some of the Network Elements that could participate in the TMN. This Recommendation standardised most of the contents of ‘Call Detail Records’ (CDRs) which until then had been proprietary or locally standardised. These ‘Call Detail Records’ have a number of uses, but when used specifically for accounting purposes they ought to be called ‘Usage Metering Records’, as in X.742 [Q.82598]. Nevertheless, the more general term is commonly used in connection with telecommunications accounting.

Operations Systems supporting billing commonly have their billing functionality subsumed into proprietary ‘Customer Care and Billing’ (CCB) systems. These systems rely on ‘Service Transaction Records’, each of which can be formed by aggregating several CDRs (for example by combining the CDRs from the exchanges involved in a call). The transformation of CDRs into service transaction records is usually carried out by mediation systems. These systems can also perform basic rating by selecting the tariff which applies to a particular service transaction. Mediation functionality may be present in the switch, provided separately or as part of the CCB system. Service transaction records can then be related to a specific subscriber’s ‘service transaction’, priced, and used as a basis for billing.

Figure 3.10 shows an overview information model for common telecommunications accounting constructs.

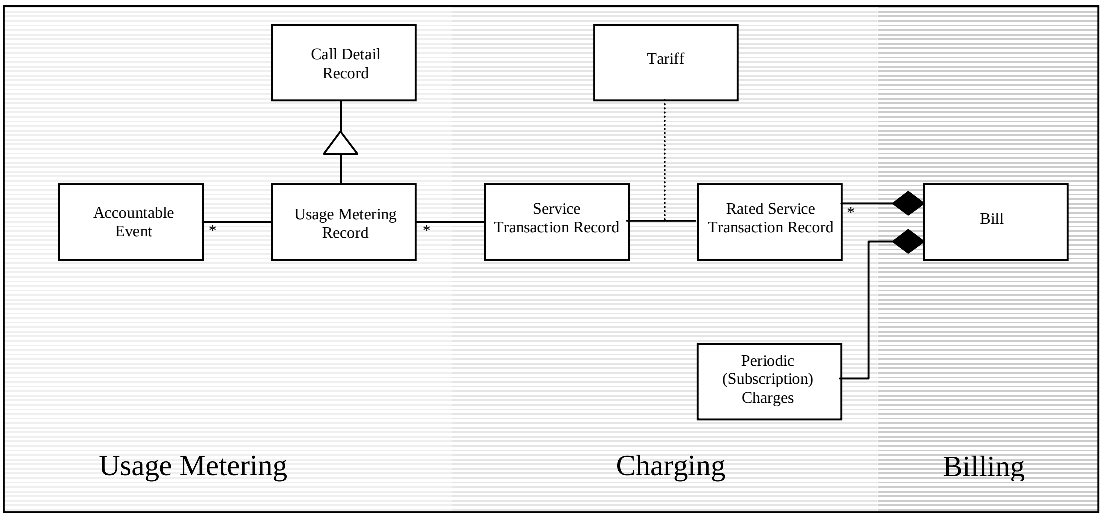


Figure 3.10, overview information model for telecommunications accounting

Research and standardisation effort in telecommunications accounting has tended to concentrate on facilitating accounting between the increasing numbers of service providers in deregulated markets [P407D197], [Bleakley97]. Subscribers can normally be charged for service transactions based on the CDR information available to their own service provider, even if they use the facilities of other service providers in the course of their transaction (e.g., for international calls, the relatively high rates used are almost guaranteed to cover the other service provider’s wholesale interconnect charges!). Other service providers involved in transactions can seek remuneration for the use of their facilities via bi-lateral agreements or increasingly through a ‘clearing centre’ [ETSITR101] or ‘clearing house’ [P407D197]. Such ‘inter-domain’ accounting (i.e. between service provider domains) normally involves the bulk transfer of CDRs, although inter-domain accounting between national service providers can sometimes be carried out on a per service transaction (i.e. call) basis [Mäkelä99].

3.4.3 Accounting in IN.

As discussed in section 3.3.1.4, IN standardisation concentrated on enabling switch vendor independence for service providers, leaving operations issues to be dealt with using proprietary, or latterly, TMN-based management systems. It was recognised that accounting was fundamental to actual service deployment, no matter how quickly IN allowed services to be developed [Bauer88], but separate accounting mechanisms were thought unnecessary as IN services came to be regarded as having a similar charging basis to other telecommunications services [ETSITR101]. As the signalling network became increasingly used to support IN services, it was recognised that some mechanisms were needed to account for this use, but as yet standards have suggested only three charging options: no charge, flat rate charges and volume based charges [Mäkelä99], none of which allows charging based on service functionality.

IN supports alternative charging models for services like ‘Freephone’ and premium rate information services [Schulz92]. IN CS-1 defined a functional component (i.e. a ‘Service Independent Building Block’ in CS-1) which could be used to determine the “special charging treatment for a call” [Duran92]. Accounting mechanisms for antecedent (i.e. pre-IN) services could be used for these calls with few adaptations [Collet92], [Pezzutti92], in the main because calls were still charged based on duration [Lappin96], albeit at higher rates! Alternative charging models allowed the originating subscriber charge (if any) to be divided between the (network) service provider and the ‘value-added service’ (VAS) or ‘information service’ provider, who would otherwise have been regarded as the terminating subscriber. (‘Freephone’ calls resulted in a charge to the information service provider).

An IN functional component supporting calling card functionality for other services was also specified [Ranasinghe94]. Calling card services allow calls to be charged to a subscriber account identified by an input number and a PIN, rather than relying on the call’s originating directory number (DN) to identify the subscriber to charge. The ‘Service Independent Building Block’ had “the usage charge directed to a particular account which may be different to the default one” [Ranasinghe94].

Proprietary research on inter-domain aspects of IN service accounting has defined a number of functional areas dealing with the periodic transfer of charging information, querying for a specific subscriber’s charges and revenue settlement between service providers [P226D495]. A number of concepts were defined, including ‘Service Detailed Records’ that corresponded to Call Detail Records for IN service invocations (these might have included e.g. alternative subscriber identification information for calling card services, or the subscriber to charge for call redirection). These could then be combined with CDRs to form ‘Service Transaction Records’ and rated to produce ‘Service Charging Records’. Such records, when aggregated into an ‘IP [Intelligent Peripheral] service account file’, were to be the basis of the charging information exchanged between providers. Several of these concepts found their way into Recommendation Q.825 [Q.82598] which is detailed in the next section.

“However elaborate the call processing, services cannot be deployed until billing and recording arrangements are made” [Bauer88]. While recognised at the outset, each generation of IN services dictated changes in what were either proprietary systems, which were frequently inflexible, or systems built to a separate set of standards that did not see the ability to charge for as yet unknown new services as an important requirement. IN services, especially both the alternative charging and calling card services, wrought changes to accounting systems. Not only did these changes have potential to delay service introduction, perhaps more than any other operations systems issue, but implementation errors could hold serious financial consequences.

3.4.4 Accounting in TMN

Accounting in TMN is based on the concepts and management functions defined in the OSI systems management recommendations. Several common concepts were introduced earlier in section 3.4.2, this section details additional concepts defined in the OSI model and also details management concepts and functions specific to TMN based telecommunications accounting.

Figure 3.11 shows the accounting model described in X.742 [X.74298] in UML. The 'Usage Metering Function for Accounting Purposes' defined therein is mainly concerned with the first stage of the typical three stage accounting process. The folowing sections briefly describe how the usage metering record (i.e. the CDR) can be produced in a system adhering to this standard.

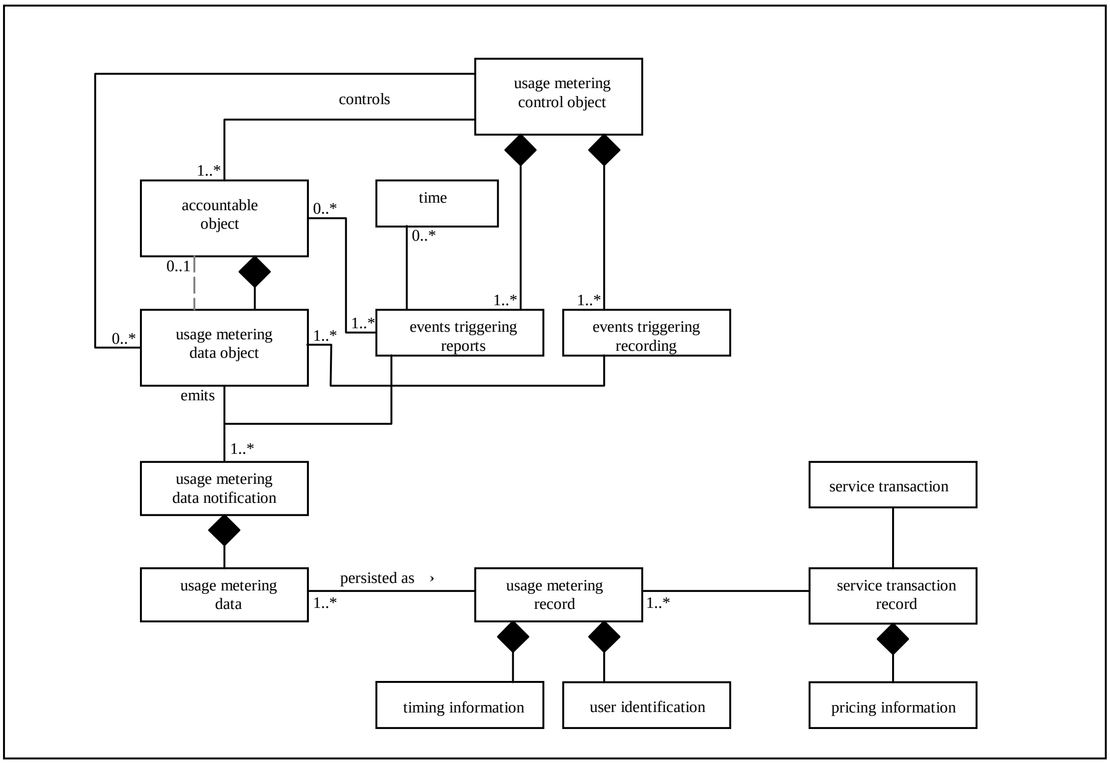


Figure 3.11, X.742 concepts

The principal managed object in the usage metering function is the 'Usage Metering Control Object' which controls the gathering and reporting of "resource utilization data" from 'Accountable Objects' which represent resources for which usage data are to be maintained.

The control object can "support and reference several accountable objects". It contains a table of 'reporting triggers' corresponding to events that can be generated by accountable objects or a timer. When an accountable object, or the timer, emits an event corresponding to an entry in this table, this in turn causes the associated 'Usage Metering Data' managed object to emit a notification containing the relevant usage data. The events defined in the 'reporting triggers' table are thus accountable events.

If there is a need to control the actual metering process, the control object can also contain 'recording triggers' to specify when measurements should be updated. The control object also specifies what usage data is relevant when a notification is generated for a particular accountable object. The attributes for the units of usage are not dictated by the standard, instead they "must be provided in a specialisation".

When the "accountable object exists solely for the purpose of accounting" the standard dictates that the "usage metering data" capability be included in the accountable object rather than be implemented separately (this is the containment relationship shown), otherwise a 'Usage Metering Data' managed object can measure usage for an associated 'Accountable Object' (this possibility is shown by the shaded numerated association). The standard also allows that the functionality of the 'Usage Metering Control' managed object be subsumed into the 'Accountable Object', although this is not shown on the diagram for the sake of clarity.

The notification emitted by the 'Usage Metering Data' managed object contains the 'Usage Metering Data' information necessary to form a 'Usage Metering Record'. These data "represent the accounted use of a resource". They are represented as a "series of basic information blocks" corresponding to accountable events. The potential contents of each block is shown in Table 3.1

Registration	Detection of service requestor (e.g. off hook)
Request	Input from requestor (e.g. dialled/keyed information)
Accept	Distant end responding (e.g. answer)
Complete	Completion of provided service (e.g. hang up)
Corresponding	Correlation information used to match records when creating 'Service Transaction Records' (STR)
Bulk	Non-event related usage measurement
Interruption	Abnormal events (e.g. clock adjustment)

Table 3.1, Potential accountable events in X.742 (clauses 8.2.3.1-7 [X.74298]

Most of these events are related to traditional telephony connection set-up and tear-down, although it should be borne in mind that each exchange involved can create a 'Usage

Metering Record' with the data supplied. These can then be combined to create a 'Service Transaction Record' by using the 'Corresponding' block that is contained (exactly once) in the series of usage data blocks (an ASN.1 'SEQUENCE OF CHOICE', expressed more familiarly as a sequence of a union type).

While for telephony service the creation of the actual 'Service Transaction Record' (STR) seems well supported, the model offers little support for newer multiparty, multimedia services, and even less for connectionless services. The "detailed mechanism" for the creation of STRs is not defined in the standard, instead it is "left open for specializers to specify". This is not surprising, given that the standard fails to properly define the notion of the 'Service Transaction' itself, making do with a dictum that it be "used in its usual English meaning to denote things like a telephone call or the sending of an electronic mail message" [X.74298].

While the lack of exactitude in the definition of the 'Service Transaction' may seem trivial at first, it should appear less so when it is realised that the 'Service Transaction Record' is the basic data upon which subsequent tariffs must be applied in charging algorithms. If the creation of the record itself is service specific, it cannot but discourage a common charging mechanism. (In section 3.3.3.3 we saw how the TINA Service Architecture defined a consistent context for service interactions, the session. This report later reveals how this concept can be applied for accounting purposes.)

The remainder of this section describes telecommunications specific standards describing the creation of 'Usage Metering Records' (or less specifically 'Call Detail Records') for subtypes of telephony service. It then describes the typical elements in a TMN based accounting system before discussing TMN based functional specifications.

TMN Recommendation M.3400 describes TMN management functions that can be applied to different network technologies to support the TMN 'management services' listed and described in TMN Recommendation M.3200. One of these management services is "Tariff and Charging Administration", specified to contain two functional areas ('Management Service Components') namely 'Tariff Administration' and 'Call Detail Data Collection Management'.

Tariff administration was to cover the creation and deletion of tariffs that corresponded to "a certain service, origin and destination" [M.340092] and also to facilitate changing the tariffs that TMN NEs should apply. Tariffs could be created to cover certain times of the day and of the week; these correspond to the ‘peak-load’ pricing discussed in section 2.2.2. As yet there has been no specifications of this functional area.

The second functional area was defined for “analogue, digital and Integrated Services Digital Network (ISDN)” services in Recommendation Q.825 [Q.82598]. This standard defines a specialisation of the functions defined in X.742 for Signalling System No. 7 (SS7) signalling networks (used as a transport network for IN services), defining three new functions to facilitate ‘Call Detail Recording’ on Network Elements supporting the Q3 interface (in this case via a mapping of CMIP to SS7). These functions are part of the specification of the ‘Call detail Data Collection Management’ functional area. They support the generation and subsequent transfer of standardised ‘Call Detail Records’ (CDRs) to TMN based Operations Systems (OSs).

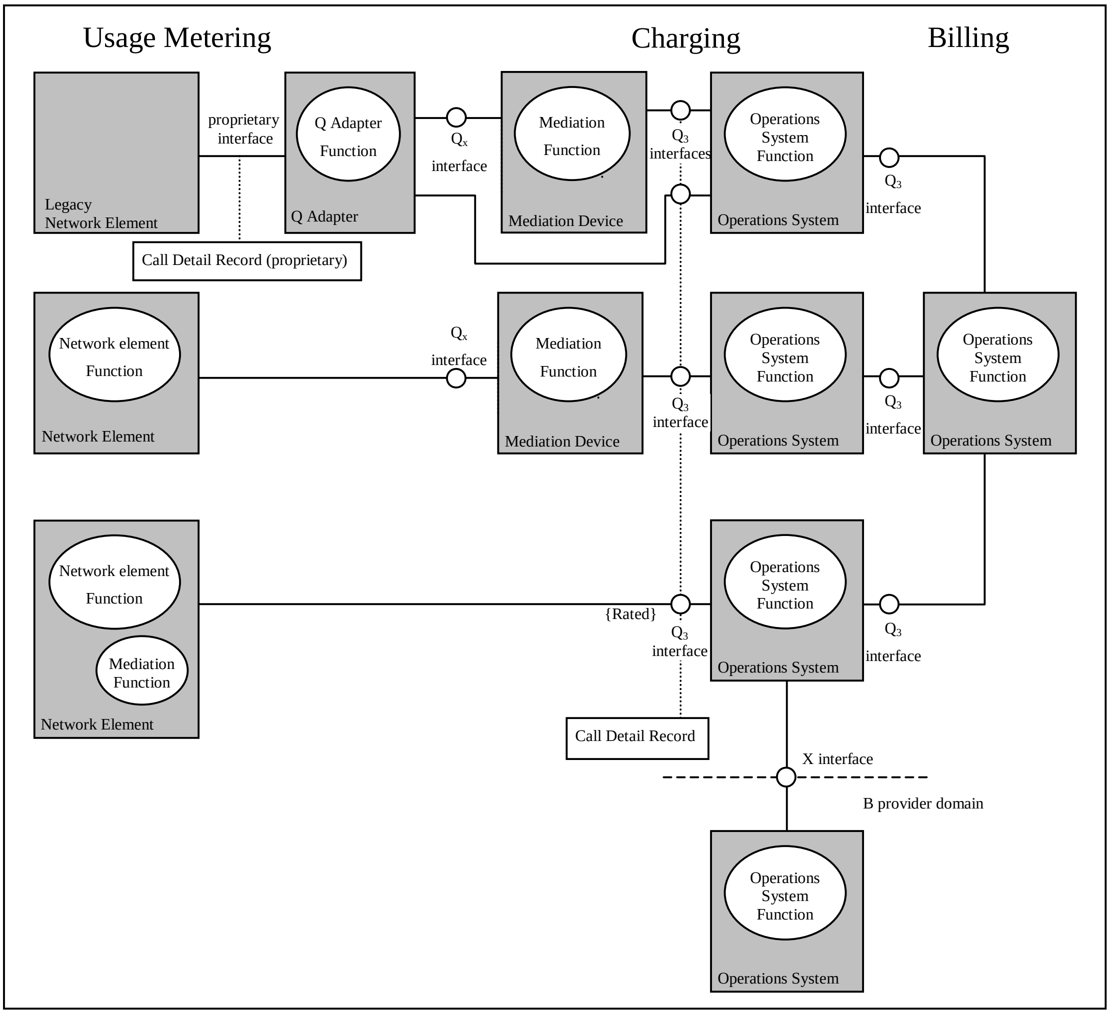


Figure 3.12, Sample TMN accounting configurations

Figure 3.12 shows how TMN nodes can interact to facilitate telecommunications accounting. It shows a number of scenarios for the interconnection of switches producing standard and non-standard CDRs and charging and billing operations systems.

The topmost series of nodes considers a scenario where a switch produces CDRs in a proprietary format. In this case, as in all cases where the TMN must interconnect with non-TMN systems, a Q-Adapter function is used to transfer (e.g. using FTP or FTAM) and convert these proprietary CDRs (e.g. in an ASCII flat file format) into a form suitable for further processing within the TMN. In some cases this is all the processing that is necessary before they are passed to a billing OS, but other circumstances may require the use of a Mediation Function which performs further processing on the records, e.g. translating between date formats or thresholding. CDRs are often aggregated to form 'Service Transaction Records' (STR) on a per-service basis at this stage; charging information can also be added. The Q-Adapter Function and Mediation Function are frequently combined in commercial Mediation Devices which integrate capabilities offering data collection from several network element types and its translation, formatting and eventual forwarding to an Operations System where it can be used for billing. [Hssworld99], [Comptel.99].

The second series of nodes considers a similar scenario where the CDRs are generated and can be collected in a standardised manner; in this case a Q-Adapter function is not necessary. The third series shows the scenario presented in [Q.82598] where the Network Element is itself capable of producing CDRs for the billing Operations Systems. In some cases these CDRs are already rated and aggregated based on a "Call Identification Number" which may or may not be globally unique or have only "local significance". This means that the identification number does not necessarily provide a means for CDR aggregation between switches and heuristic aggregation methods based on in-going or out-going trunk groups or the called number may still be necessary.

The Operations Systems may co-operate with peer Operations Systems in other provider domains to facilitate inter-domain accounting, they may also be used by an Operations System at a higher level in the 'layered functional hierarchy' (see Figure 3.4). Typically inter-domain accounting occurs at the service level, while interfaces offering similar functionality (but provided by a Q_3 interface not an X interface) allow operations systems in the business layer to access the information necessary for billing. Operations Systems supporting billing functionality in the business layer are often subsumed into commercial 'Customer Care and Billing' software offerings.

As stated previously, the TMN standardisation effort was generally reluctant to standardise the functionality of nodes lest it constrain market offerings [Glitho95]; this general principle also applied in the realm of accounting management. Nevertheless, some parties, outside the national standards bodies that defined the TMN, found it useful to define just such functionality to aid specification and, perhaps ultimately, software re-use.

The RACE Netman project defined such a “Common Functional Specification” for “accounting management services in TMN” [Netman94]. This outlines basic principles that it felt ought to apply when charging for telecommunications services. The guidelines suggested objective and flexible charging methods independent of the “specific teleservice”. It suggested service independence be supported either by resource based usage information (“the real number of cells transferred by the network”) or on parameters negotiated during connection set-up, based on the presence of a “policing” (e.g. traffic shaping) function, or a combination of both. The limitations of these non-price rationing methods were discussed in the previous chapter.

3.4.5 Accounting in TINA

This section considers the allowances made for accounting within various releases of the TINA architecture.

3.4.5.1 Initial model

The TINA consortium described an outline model for accounting for TINA services in their second Service Architecture document, publicly released in 1995 [TSA95]. The informational and computational models described therein subsequently formed the basis of the accounting model for this research within the Prospect project, which itself had chosen that TINA service architecture as a basis for the management of integrated services [PD22B98].

The goals of accounting within the architecture included:

- the definition of services that could manipulate resource usage information,
- support for flexibility in charging and billing and
- support for the efficient introduction of new tariffs.

Some of the requirements to support these goals included the ability to charge for resource usage based on “various units of measure” and for “various environments”. Others included security of accounting information, on-line charging, provision for the sharing of charges between service providers for ‘compound’ services and between subscribers, and enabling bill enquiries.

Figure 3.13 shows a UML model for TINA Accounting, based on the OMT information model given in [TSA95]. The central concepts in this model are the ‘Usage’, ‘Charge’ and ‘Bill’ which were related through two tariff types. A ‘BasicTariff’ delineated the relationship between the usage information for a resource (represented by an accountable object) and the charges for that usage. The ‘UserPlan’ could then be applied to the list of generated charges.

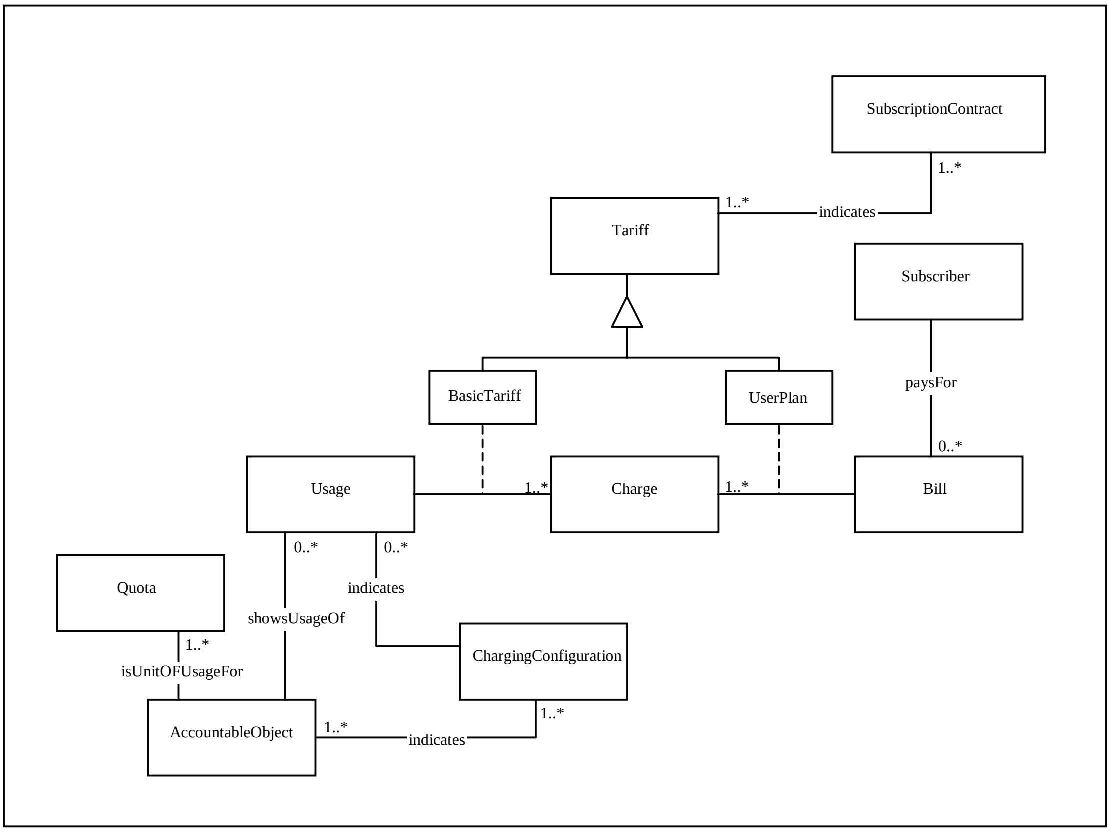


Figure 3.13, TINA Accounting Information Model (based on that in [TSA95])

Prospect, the ACTS project within which the initial part of this research was carried out (see section 5.2.1), in the absence of further detailed specifications [PD4B98], initially modelled the ‘BasicTariff’ as an IDL structure that could contain four service independent, i.e. generic service session related, ‘accountable events’ and their charges in schedule form. Service dependent accountable events and their charges were subsequently added. (A sample Prospect ‘BasicTariff’ is shown in Figure 5.17 and a description of its role in charging is discussed in sections 5.2.3.3 and 5.2.3.4). The ‘UserPlan’ was modelled as a structure instantiated to contain a standing charge, corresponding to a rental charge, and a discretionary discount percentage for the aggregated charges.

The model contains the concept of an accountable object, a similar concept to that described in the OSI model in section 3.4.4, and a quota object which allowed different units of usage for an accountable object, e.g. a file size, a time period or a bandwidth measure. (The quota object thus plays one of the roles of the ‘Usage Metering Control’ object described in the OSI model). The charging configuration was used to indicate the recipients of the charge for resource usage. In Prospect, the atomic events of the service session meant that measurement units were not appropriate, rendering the implementation of a quota object superfluous. The availability of user and subscriber identification information for the service level events also meant that the implementation of a charge configuration object was unnecessary.

The subscriber and subscription contract objects were modelled as part of the subscription component within Prospect. The accounting component selected the tariffs to apply based on identifiers listed in subscribers' contracts managed therein.

3.4.5.2 TINA 'Accounting Management Architecture'

The TINA consortium publicly released its 'Accounting Management Architecture' specification in late 1999. Its contents were previously briefly outlined in a publicly available document [Hamada96]. After its initial specification, the accounting architecture was not maintained separately [TSMWG99]; its concepts were subsumed into the TINA Network Resource Architecture [TNRA97] and the TINA Service Component Specifications [TSCS97].

The architecture defined two new concepts for use in TINA accounting; the 'Service Transaction' and the 'Accounting Management Context'. The 'Service Transaction' was to fulfil a similar role to the 'session' concept defined in the service architecture, except it was to provide a context solely for service management; it provided a separate context for management activities which was not based on the service session or did it need to mirror its lifecycle in any way. The service transaction was based on 'Transaction Processing' concepts and was concerned only with a single user's interaction with the resources supporting a service. It was intended to carry "FCAPS service management functions whose behaviour is dictated by respective contexts" [Hamada96], i.e. within different domains.

The 'Accounting Management context' was specified, as a part of a larger management context, to dictate the behaviour of a service transaction with respect to accounting in a particular domain. It was intended to represent the "one and only Reference Point between two providers regarding their accounting management", and thus can be viewed as an accounting record suitable for interchange between providers. In the service component specification, the management context is defined to contain a property list that IDL comments specify should contain properties "such as tariffId, billing options, recovery options etc" [TSCS97], then stated that the set of properties making up a management schema would be defined separately in a respective management architecture!

The separate accounting management architecture was not greeted enthusiastically, with one reviewer commenting in an annex: "…we have all that we need in the architecture in the way of management at the moment. Why do we need another TINA accounting architecture?" [TAA96]. It was also suggested that the "approach [was] unnecessarily complicated". Evidence from TINA validation projects seemed to bear this out: for example, the ACTS project VITAL still used service level accounting components based on earlier service architecture concepts (although only the 'UMData' and 'UMLog' components were ever implemented; the tariffing, charging and billing activities were "emulated", so no service level charging ever actually occurred [VD1197]).

While the concepts of the service transaction and accounting management context were greeted less than enthusiastically by commentators and implementers alike, several of the lower level component specifications provided a useful basis for purely network based accounting mechanisms used in billing for 'Premium IP' (IP over ATM) services in SUSIE, another ACTS project. The network based accounting from this project was used together with service level accounting components developed as part of this research in the ACTS Flowthru project (see section 5.3).

3.5 Summary and conclusions

This chapter presented several telecommunications software architectures in approximate historical order. It evaluated the possibility of each defining a generic and re-usable means of accounting for telecommunication services and then looked at the specific provision made for accounting in each architecture. The historical presentation helped to highlight the growing independence from underlying network technologies in the definition of telecommunications services.

The first telecommunications software architecture discussed was the IN, the 'Intelligent Network'. Its history and objectives were outlined initially, before an analysis which determined that it was technology specific; that its detailed standards tended to stifle service innovations; that its proprietary service creation environments precluded true software interoperability, and therefore re-use; that it was limited by contemporary software technology; and that it had a propensity to neglect operations issues. Nevertheless the IN initiative did help to create a commodity market for telecommunications equipment while promoting a physically based software distribution that established a basis for the separation between services and their delivery. Accounting within the IN was discussed separately in section 3.4.3; this showed that it initially dealt with accounting in a service specific - and generally proprietary - way, latterly turning to TMN based operations support systems. The analysis then considered IN's support for alternative charging models and component based support for 'calling card' services before examining how the IN attempted to facilitate inter-domain accounting guided by TMN principles. It showed that the general criticisms of the initiative also applied to its treatment of accounting for telecommunications services.

The TMN, an operations system software architecture, was then discussed. Its history and objectives were outlined, before an analysis which determined that it lacked support for service creation and that its reluctance to specify functionality, with accompanying software technology limitations, discouraged true software re-use. However, the TMN concept of management layering was regarded as a useful concept to help to partition and reduce the complexity of telecommunications operations systems. Accounting within TMN was considered in section 3.4.4; the general OSI accounting model was described, along with a telecommunications specific accounting model. These were shown to be biased towards connection oriented services, and in particular telephony services, and there was also criticism of the lack of a standard basis for charging which precluded a service independent charging mechanism. Finally, the analysis considered a suggested functional specification for TMN based accounting systems and TMN's support for inter-domain accounting.

The last architecture discussed was TINA, the 'Telecommunications Information Networking Architecture'. This was analysed with reference to the perceived shortcomings and benefits of the previous architectures. TINA was shown to have recognised the importance of software re-use, defining an architecture embodying functionality separations and management layering that encouraged the specification of re-usable service components. The origin of one of the most important architectural concepts – the session – was outlined, along with the importance of the service session in facilitating a network independent view of service use, seen as a suitable basis for service independent accounting. While the architecture recognised the need for software re-use, it was shown that its support for such re-use, in particular run-time re-use, was inadequate. The architecture was also shown to have a bias towards connection oriented services. Accounting within TINA was considered in sections 3.4.5.1 and 3.4.5.2. Section 3.4.5.1 considered the TINA accounting model used as a basis for this research within the Prospect project (this is evaluated, within the context of this research, in section 5.2.5). Section 3.4.5.2 considered a later TINA network level accounting model which was based on transaction processing concepts.

Overall the analysis shows that both the IN and TMN telecommunications software architectures could not support a service independent accounting mechanism, for reasons of a bias towards telephony and software technology limitations in IN and because of the absence of service creation concepts, and the lack of a service independent charging basis, in TMN. Concepts in the TINA architecture were shown to help address these shortcomings.

The TINA architecture addressed several of the requirements that might be necessary for service independence, yet the TINA accounting mechanisms are still based on resource level usage data; for example, network level parameters exist in tariff schemes. This betrays a network technology dependence in service charging mechanisms, while the services themselves are increasingly capable of being defined without regard for the underlying technology. This technology centric approach to charging fails to address the perceived value in using a service and also has repercussions on the ability to provide a service independent accounting mechanism. The traditional and contemporary concentration on producing, aggregating and interpreting network level measures for service usage has ignored the fact that a software defined service has the innate ability to produce its own usage information that can be used as a basis for charging.

4 Design of a generic accounting architecture for telecommunications services

> "Design is […] a contingent process, subject to changes brought about by conditions that come to the surface after the big decisions have been made." [Ferguson92]

4.1 Introduction

The previous chapters described several approaches to accounting for network based services. It was shown that, due to their limited application domains, they lacked the generality to be seamlessly applied to services based on different communications paradigms. It was also shown that, where service based accounting does exist, it is primarily based on the cost of using underlying resources rather than the perceived value of the service to its subscribers. This may not provide an adequate basis for charging.

From the previous chapter's description of a telecommunications service, it is apparent that such services can now be regarded as highly reliable software systems. Software system design has developed into a discipline of its own since Brooks' seminal work [Brooks75], but the importance of what he called 'conceptual integrity' remains as true to-day as it was then. Jacobson, who honed his software skills developing switch based services for the Ericsson AXE series of switches, describes an architecture as a 'common vision':

> "We can think of the architecture of a system as the common vision that all the workers (i.e., developers and other stakeholders) must agree on or at least accept." [Jacobson99]

This chapter describes a generic service accounting architecture founded upon the service and network separation fostered by the TINA Consortium. It encapsulates a series of concepts deemed necessary to account, price and charge for network based services. These services depend on telecommunication, but need not be either 'in' or 'of' the network in the traditional telecommunications taxonomy. This chapter begins by outlining the primary design influences, then specifies the requirements for a generic networked service accounting architecture before outlining the primary architectural concerns and the technical difficulties these pose. It then describes the stages in the evolution of a design to meet these requirements.

4.2 Design influences

An architecture to capture the concepts of pricing and charging for network based services, be they traditional telecommunications services or Internet based information services, must be guided by the market for such services. A business model helps by identifying the market stakeholders and the roles they play in creating, providing and using a service; the TINA Consortium business model is one such model which, although biased towards connection oriented connectivity providers [PD14b98], nevertheless provides useful foundational roles.

The initial TINA business model is shown in Figure 4.1. TINA roles are shown by the shaded boxes. Relationships between these roles are shown by the connecting lines; they are called ‘reference points’ in the TINA argot. The reference points are outlined in Table 4.1. It should be noted that domains are administrative, not technological, and that in some cases the roles can be played by the same stakeholders, e.g. a retailer offering services of its own. The following sections outline the influences on a generic accounting architecture for network based services that can be seen within the context of this model.

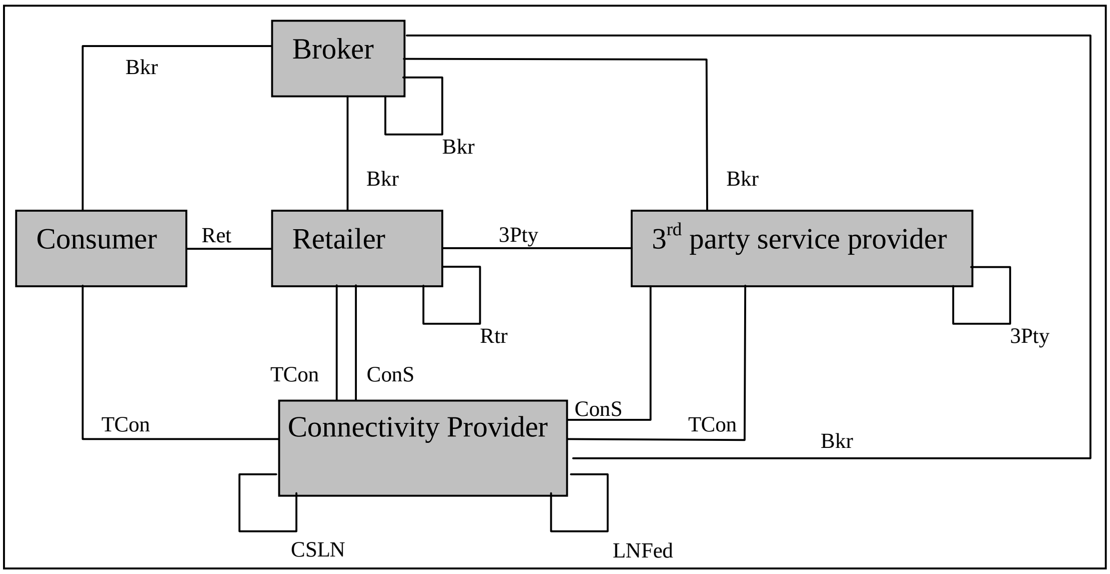


Figure 4.1,The TINA ‘key business areas’ and their relationships[TINABM97]

Ret	Retailer inter-domain reference point	ConS	Connectivity service inter-domain reference point.
Bkr	Broker inter-domain reference point	TCon	Terminal connection inter-domain reference point
3Pty	3rd Party inter-domain reference point	LNFed	Layer network federation inter-domain reference point
Rtr	Retailer-to-retailer inter-domain reference point	CSLN	Client-Server layer network inter-domain reference point

Table 4.1, Tina reference points

4.2.1 DPE based software architectures

Advanced telecommunications services were made possible with the advent of stored program control switches in the 1960's. New services began to be defined in terms of the facilities switches offered to these programs, and gradually manufacturers offered switches with more facilities. These facilities were, and still are, the basis of 'switch based services'. The 'Intelligent Network' (IN) initiative of the 1980's (described in detail in section 3.3.1) offered a distributed software architecture for telecommunications services to mitigate the expense of provisioning switches with the capabilities required by sharing service resources between switches. IN defines services using Abstract Syntax Notation 1 (ASN.1) to be invoked using the OSI remote operations service (ROS), but does not address issues arising from the distribution of the actual service application software [Barr93] and does not benefit from the distribution transparency, application services and potential development rapidity afforded by a DPE; nevertheless, most telephony services are still based on the capabilities offered by the IN.

While the main role of the IN was to separate the provision of subscriber services from the provision of switches, the role of the TMN was to divorce the provision of Operations Systems (OS) from the provision of switches [Kockelmans95] (called Network Elements (NE) in the TMN terminology) to facilitate telecommunication deregulation [Davison99]. Again the architecture was driven by market realities. TMN, primarily established to facilitate element management, moved to provide service management on a similar basis; this was fraught with architectural problems, mainly due to the lack of a sufficient distributed processing environment (DPE) specification [Pavon96], [Prepare96].

The Telecommunications Information Networking Architecture Consortium (TINA-C) was founded in 1993 against the background of what it thought were expensive and unnecessary separations between service operations, management and control and largely unsuccessful attempts at service creation by the TMN community [Appledorn93] and service management by the IN community. The TINA Consortium defined an architecture for 'distributed telecommunications software applications' [TINAB99], divided into a DPE based computing architecture which supported a service architecture and a network architecture.

While TINA has been criticised for its bias towards connection-oriented communications in its network architecture [Lewis97], its computing and service architecture concepts were successfully applied to Internet based services in the Prospect project [PD14b98]. This was facilitated by the TINA separation of the service and network which allowed services to be developed independently of the underlying communications technology. The service and network separation also introduced the session concept, the origins and specialisation of which were discussed in section 3.3.3.3.

The market led introduction of a DPE based service architecture, such as TINA-C's, allows true software based service creation and provision and is a necessary foundation for the specification of a generic accounting architecture.

4.2.2 The regulatory environment

The regulatory changes brought about by, amongst others, the United States' federal government and the European Commission in the 1980's and 1990's [Bangemann97], ushered in a new era of competition for network based service providers. For example, the European Commission 'Services Directive' of 1990 [90/388/EC] removed exclusive rights for the supply of value added services and data services from member states' traditional operators. This allowed other providers to offer services on their networks; this reality is reflected by the presence of the separate third party service provider in the TINA business model (Figure 4.1 above).

Operators sought to rapidly create new services to differentiate themselves from their competitors, but were soon faced with the limitations of the software architecture used to construct these services [Zave93]. For both operators and 'service logic providers' [TINABM97] (i.e. service creators) the prospect of diminishing returns based on technological complication and software incompatibilities led to an increased awareness of a need for a standard framework for new services. The traditional telecommunications standards body, the ITU, with its four year study periods, was deemed to be too slow to react to the new market realities. As a consequence of the perceived confusion and the need for a faster standardisation process, the TINA Consortium was born in 1993 with the motto "A co-operative solution for a competitive world", joining many other industry forums, such as the Tele-Management Forum (TMF), as *ad hoc* standards bodies.

The regulatory environment and the new market realities imposed by resultant competition lead both operators and third party service providers to create and provision services at an ever increasing rate. These services may be provided by someone other than the network provider and need a pricing and charging architecture that supports their rapid development and heterogeneous nature.

4.2.3 The Internet

No one, particularly if they are involved in the telecommunications industry, can ignore the phenomenon of the Internet. The Internet is application blind; it simply moves packets without any regard for their content. This has the obvious drawback that any user can attest to; communication has the potential to be cripplingly slow, as the network resources are shared in a non-discriminatory way (although there have been attempts to change this by at least one router manufacturer, CISCO Systems [YoungJS99]).

The explosive growth of Internet subscriber numbers demonstrates that this drawback is more than balanced by the ease of introduction and adoption of new services [Odlyzko98c]. These new services created need not depend directly on any underlying communications mechanism, on the network architecture or an intelligent network; instead the intelligence is removed to the edges of the network; to smart terminals, i.e. computers [Isenberg98]. New service introduction and service upgrades are no longer the concern of the network operator; service users must maintain the service themselves.

The responsibility for transparent service upgrades is a fundamental part of service provisioning and management that the experienced Internet user has accepted. Given the effort required, new, relatively naïve, and experienced but indolent users all too quickly realise the potential benefits of invisible, stealth based, service upgrades. This points to a potentially huge market of users willing to delegate service management to a service provider which is, after all, part of their traditional role.

Traditional telecommunications service providers can build on their experience to market fully managed services. The use of these services will need to be priced, accounted and charged for; here the service and network separation mooted by TINA can allow their service architecture to be applied to Internet services and subsequently allow them to be charged for by any TINA based mechanism. It also points to the need for a clear service based charging scheme independent of the underlying network technology, particularly if the use of the network is ‘free’.

4.2.4 Network economics

As detailed in a previous chapter, where once the industrial economy was driven by supply side economies of scale, now it is driven by the economics of networks. Here the concept of positive feedback dominates; a product’s success begets more success. Traditionally positive feedback is referred to by economists as ‘network effects’, ‘network externalities’ or ‘demand-side economies of scale’ and is generally treated as aberrant behaviour because it challenges the rational basis of neo-classical economic thought. Unfortunately these network externalities are a defining feature of information industries; the main problem facing a new product is igniting enough initial interest in the marketplace to create the virtuous circle of the positive feedback loop [Shapiro99].

The value of positive feedback applies to telecommunications services as to any other information product; the more people that use a service, the more people will want to use that service. The recent growth of mobile phone adoption in Europe is a recent example of network effects in action for the telecommunications industry.

At the moment, Internet based information services are either funded through flat-fee based subscription or advertising, or are used to add value to an existing information resource as part of a product bundle; this is primarily due to the lack of suitable accounting or payment mechanisms for Internet based information services.

Telecommunications services and information services have similar cost bases and both benefit (or suffer!) from network effects; this points to a pricing scheme based on their commonalties. Pricing based on network externalities requires a good market model and some ability to change prices based on service demand; in the case of a separate accounting system, this requires some form of explicit demand indication from the service. Changing prices based on demand also requires that there be defined minimum and maximum charges (for regulatory purposes) and that users be informed of service costs in real time.

4.3 Requirements for a generic accounting architecture

This section details technical requirements on a generic accounting architecture guided by the market based influences above. Based on these influences, an accounting architecture must support:

- re-usable components:
  - the location and access and transparencies distributed processing environments offer enable component based software re-use,
- easily defined and extensible tariffs:
  - this is the primary means to reduce the time to market for new services within an accounting architecture, although time to market is also reduced by appropriate component based service software re-use,

- service based charging:
  - charging can be based on perceived utility, not just resource cost recovery; estimating perceived utility also enables demand based pricing. Charging in this way grants network technology independence,
- adaptive, per-user pricing:
  - an accounting architecture should support flexible pricing mechanisms that can adapt to current demand. The most flexible pricing mechanism is per user pricing, which can enable perfect price discrimination.

The market influences given in the previous section shaped these architectural requirements, although there is not always a one to one mapping between them; some requirements exhibit similar influences in their application - for example service based charging enables demand based pricing.

In designing an architecture, the general process followed the iterative approach detailed in [Mowbray95], in that there was an initial architectural composition refined through prototype experience. Some of the requirements listed here were present for the first architectural prototype, others were perceived in later stages; the order of these requirements reflects the experience gained in defining intermediate architectures.

4.3.1 Re-usable components

Among any architecture’s primary aims should be the fostering of software re-use [Jacobson99]. Szyperski maintains that true software re-use can only happen through the use of software components, where the author is quite specific in what is meant by a software component:

> “… software components are binary units of independent production, acquisition and deployment that interact to form a functioning system.” [Szyperski97]

Thus, Szyperski maintains, the use of such executable components should improve software quality, and therefore reliability, and also support rapid development, leading to a shorter time to market. Both these qualities of component based software fit comfortably with the demands of new telecommunications services for reliability and speed to market, and can be regarded as a fundamental concept in the definition of telecommunications services [Li94].

Until relatively recently, components were not possible outside quite narrow application specific areas, but the adoption of CORBA as a *de facto* standard by the computer industry has enabled application services, facilities and application-domain-specific standards to be defined within the OMG's OMA architecture. Initially reluctant to specify 'on-the-wire' protocols, the group found it necessary, for reasons of interoperability, to specify such a protocol for its second architectural release in 1995. The CORBA 2.0 architecture defined a generic inter-orb operability protocol and a standard mapping to IP; the Internet Inter-Orb Protocol or IIOP. With IIOP a true component software market could begin. While the OMG is still continuing technological standardisation efforts [OMG99-02-05] to help promote a market for component software, others believe fundamental reward schemes need to be in place before such a marketplace becomes a reality [Cox96].

The TINA Consortium recognised the advantages a distributed processing environment offered the inherently distributed nature of telecommunications services. Initially defining their own DPE based on ANSAware [APM93] [Herbert94] the consortium then adopted the OMG's CORBA architecture, while adding some telecommunications specific stream interfaces and interface groups to the IDL, producing 'Object description language' (ODL) [Barr93]. The proposed accounting architecture is applicable to networked services defined using the DPE based TINA Service Architecture [TSA95] and builds on it to offer a component based solution to ease the introduction of new services.

4.3.2 Easily defined and extensible tariffs

Newly deregulated telecommunications markets require a regulator to ensure that the market incumbent does not abuse its position of market dominance. Part of such regulation is an insistence on tariff publication to ensure predatory pricing or other pricing practices encouraging 'lock-in' (effectively forcing a customer to stay with a specific provider) do not occur.

As telecommunications markets become more open, there comes an increasing need for a clear tariff definition to enable automatic comparison of tariffs; this would allow potential service subscribers to choose -perhaps dynamically- different service providers for different services. This feature would be an extension of the current service-provider pre-selection alternatives available to-day in European markets, and would be similar to the choice available in the U.S. market, but could be automated in a similar way to the operation of least cost routing carried out presently by some PABX systems for long distance service.

New telecommunications services are traditionally built on top of older defined services [Griffeth93]; while this leads to problems in itself (like those of 'feature interactions' [Cameron93b]) it demonstrates the need to develop tariffs in an extensible manner to enable easier introduction of new services. Extensible tariffs would also allow subscribers to dynamically extend existing service options and choose new services, while making cost implications clearer

4.3.3 Service based charging

Traditional telecommunications tariffs were designed with the premise of scarce resources in mind; they served to dampen demand during peak hours because the single application networks (i.e. telephony) were designed to cope with a certain peak traffic (‘busy hour’). Scarcity of bandwidth almost followed from the price restrictions in the old telecommunications monopoly model:

> “…we see that the regulation of profits, or constraints on the pricing system adopted may have the effect of reducing the financial incentive to provide a telephone system free of congestion.” [Littlechild79]

In the absence of monopoly, prices cannot be kept artificially high to stifle demand and the provisioning of new bandwidth becomes necessary. As detailed in the previous chapters, the telecommunications industry need no longer base its pricing on the premise of scarcity of bandwidth; in highly competitive markets the reality demonstrates this [Cairncross95].

Spurred initially by this demand, technology has driven world-wide decreases in the price of bandwidth. Once a network is provisioned, service providers have limited options for excess bandwidth; they can try to sell it at wholesale rates (e.g. on-line auctions) or they can stimulate demand to recoup their sunken costs. This extra demand comes at almost zero marginal cost, e.g. an extra phone call made when the infrastructure is already in place costs essentially nothing to provide.

Stimulating demand amounts to operators encouraging more subscribers to use new services on their networks. In a world with ever cheaper bandwidth [Bruno99] providers are acknowledging that new services “will become the main source of income for telecom operators” [McKinney98]. This demonstrates a requirement for charging based on services rendered rather than on resources used.

If the move towards packet based communication for traditional telecommunications services like voice telephony does indeed occur as some forecast [YoungJS99], the recovery costs for network utilisation costs will likely exceed the actual costs that the extra traffic has imposed on the network [Schenker96]; in this case there is an even greater need to base service charges on something other than network utilisation and move towards service based charging.

4.3.4 Adaptive per-user pricing

Regulatory pressures are leading to lower switching costs (i.e. between service providers) for subscribers at a time of real competition in the local loop; traditional telecommunications providers, cable television companies and even electricity companies are vying for the same subscribers. Competition is forcing service providers to introduce new services and to create loyalty schemes and subscriber specific tariffs (e.g. ‘Friends and family’) to encourage subscribers to stay with their present provider. At the same time, providers are deriving more of their revenue from their own, and third party, services to extract the maximum return from their infrastructural investments.

Offering short term subscriber specific tariffs based on a correlation of actual (or short term projected) service load with subscriber usage profiles would be advantageous to both service subscriber and provider. Subscribers could get targeted special offers based on their usage profiles with a particular service provider, while service providers would promote service utilisation, thereby increasing revenues, and encourage subscriber loyalty. This is a means of achieving perfect price discrimination by using differential pricing. If only one provider in a market provided such a scheme, this could also be a means of product differentiation.

Such a scheme requires real time adaptive pricing based on service load, i.e. demand based pricing, and also requires that prospective users know the true cost of using a service. This requires service costs to be made available to each subscriber’s user in real-time via a feedback mechanism.

4.4 A proposed architecture

The requirements detailed above dictate a DPE based, component centred, service accounting architecture that supports easily defined and extensible tariffs, service based charging, and feedback mechanisms sustaining adaptive per-user tariffs to enable real time, demand-based pricing. The following sections detail the principles that form the basis for an architecture to support these requirements. While the presence of each principle is explained in its own right, a determination of whether a service accounting system built according to these principles meets the requirements imposed in the previous section is presented in chapter 6, in the context of an overall architectural evaluation.

It is recognised that while architectures are shaped by the needs of their particular application domain, they must also pay attention to more generic technological considerations for software built as a result of their application. These technical considerations are outlined in each section when relevant.

4.4.1 Service architecture and session basis

Telecommunications service architectures were discussed in the previous chapter and their direct influences on design were detailed in section 4.2 of this chapter. The accounting architecture was designed to complement services that conform to the TINA service architecture. The use of the TINA service architecture brought the following advantages:

- the concept of a session as a context for user/service interactions,
- service specifications that facilitate component based service accounting,
- separation between the service and network, facilitating service level accounting.

The use of the TINA service architecture also allowed the use of several existing services to assist in the validation of the accounting architecture. A TINA session defines a ‘relation between entities’ that, in the case of a communications session, only considers connectivity ‘from a high level point of view’ [TINAGOT97]; in a similar manner a service based accounting architecture takes a high level view of the communications sessions used to support a service session - the communication session identifiers associated with each service session are recorded and can be used to facilitate communication based accounting if this is required.

The service session provides a stateful context for a user’s interactions with a service. The session tallies the number of parties in a multi-party session and its life cycle mirrors the actual use of the service by those parties. (A multi-party session need not necessarily have multiple users; a user can be represented by more than one party in a session, each perhaps playing different roles).

While session control is service independent, actual service usage in the context of this session is liable to be service specific. Sessions ease the burden of management by reducing the amount of contextual-information events generated by a service have to carry to be useful; a user could be participating in many service sessions simultaneously and the volume of events generated by many users could be considerable. Session based usage also enables easier service customisation and facilitates better service management by allowing service providers to see exactly what users are using, or have immediate potential to use, their services.

The session concept is a useful one for accounting purposes, especially in the case of adaptable subscriber specific tariffs, because it eases the forecasting of short term demand for service use for the same reasons it eases actual service manageability. These two benefits have often been conflated, but need to be separated to enable a generic accounting mechanism. This means that a separate service usage context is necessary for accounting purposes: this 'accounting session' should mirror the life cycle of the service session but provide a separate context for accounting events (see section 4.4.3). It was found that the TINA session concept was a required, but insufficient, basis for service based accounting; this is discussed in chapter 6.

4.4.2 Extensibility and interoperability

Extensible tariffs are necessary to allow service upgrades (and new services based on antecedent services) to be accounted for without resorting to service specific accounting mechanisms. To accommodate the requirement for tariff extensibility, the IDL data type specified as the basis of the tariff definition is founded upon list elements containing string type accounting event names, aliased in the justifiable absence of explicit subtype support in the IDL compiler.

As stated in [Mowbray95] extensibility and interoperability are often conflicting design goals; in this case, the need for extensibility - to allow accounting component re-use for different services - could cause interoperability problems in the case of a service versioning problem. To counter potential interoperability problems, the architecture requires services to publish the accounting events it is capable of generating in the 'separation of hierarchies' approach recommended in [Mowbray95]. In the accounting architecture these events, with a tariff specifying how these accounting events could be interpreted to produce charges, describe any service for accounting purposes.

4.4.3 Separation of service and accounting

Orthogonal to the TINA architecture's service and network separation, the accounting architecture aims to separate network based services and accounting for those services. As with TINA service component re-use over different network technologies, this separation is essential if re-use of accounting components for different service paradigms is to occur (e.g. digital libraries and video conferencing services).

While it is necessary for services to facilitate demand based pricing by signalling greater service load, measuring such load is strongly service specific. In this case it is up to each service to decide the basis of its load indication. This is then made available as a percentage value that can be speedily factored into charge calculations. This allows accounting services to be aware of the service load and price accordingly while being insulated from the service specific factors causing this load.

Separation is also encouraged by forcing accounting events to be defined independently of service independent and service dependent events used for other purposes, i.e. rather than defining some service events as accountable and others not. Where necessary two events (with different sinks) are generated to enforce the separation; this means that accounting can take place independently of service logic and reduces the need to change descriptions of services for accounting purposes because of service logic changes. This is not contrary to the stated TINA objective of placing management functionality within the services - the services still need to generate the accounting events - but is a solution that offers flexibility without diminishing the reusability of an accounting service.

4.4.4 Feedback

Allowing the prices for network based services to change based on service demand ought to enable both the service provider and service consumer to benefit; a service provider must view an unloaded service as uneconomical, given almost zero marginal costs for additional users, and a subscriber could see it as an opportunity to use a service cheaply.

Such pricing schemes will probably be necessary to encourage subscribers to use new and innovative services. However, to-day’s innovation could easily become to-morrow’s necessity -Internet based services are fast being viewed in this way. In the ideal case of zero service switching costs and perfect competition such a scheme might work, but tariff regulation will probably be necessary initially, and even if it is not, will probably be carried out anyway [Cuiker99]. Regulation would ensure minimum levels of return to encourage new market entrants (and discourage service subscribers from waiting out for the best service price) and set maximum levels of return to ensure subscribers are not exploited. The accounting architecture defines tariffs which define minimum and maximum prices for specific service actions. Service discrimination by consumers would probably initially focus on the steady state pricing levels for services at certain times of day as the data become available.

Changing prices dynamically mandates that service subscribers always be aware of the service costs they are incurring; this requires the use of a feedback mechanism to ensure users know service costs in real time. The accounting architecture supports such a feedback mechanism.

4.5 Architecture diagrams

The overall architecture is shown in two diagrams showing the computational and information viewpoints as per RM-ODP.

4.5.1 Computational model

Figure 4.2 shows an overview computational model. This figure uses a notation revealing operational interfaces because this clearly indicates the origin of the interactions between the components. This notation is used to describe the TINA service architecture in [TSA95] but the accounting components could be used within any service architecture based on a distributed processing environment supporting a secure usage context for service interactions, i.e. a service session.

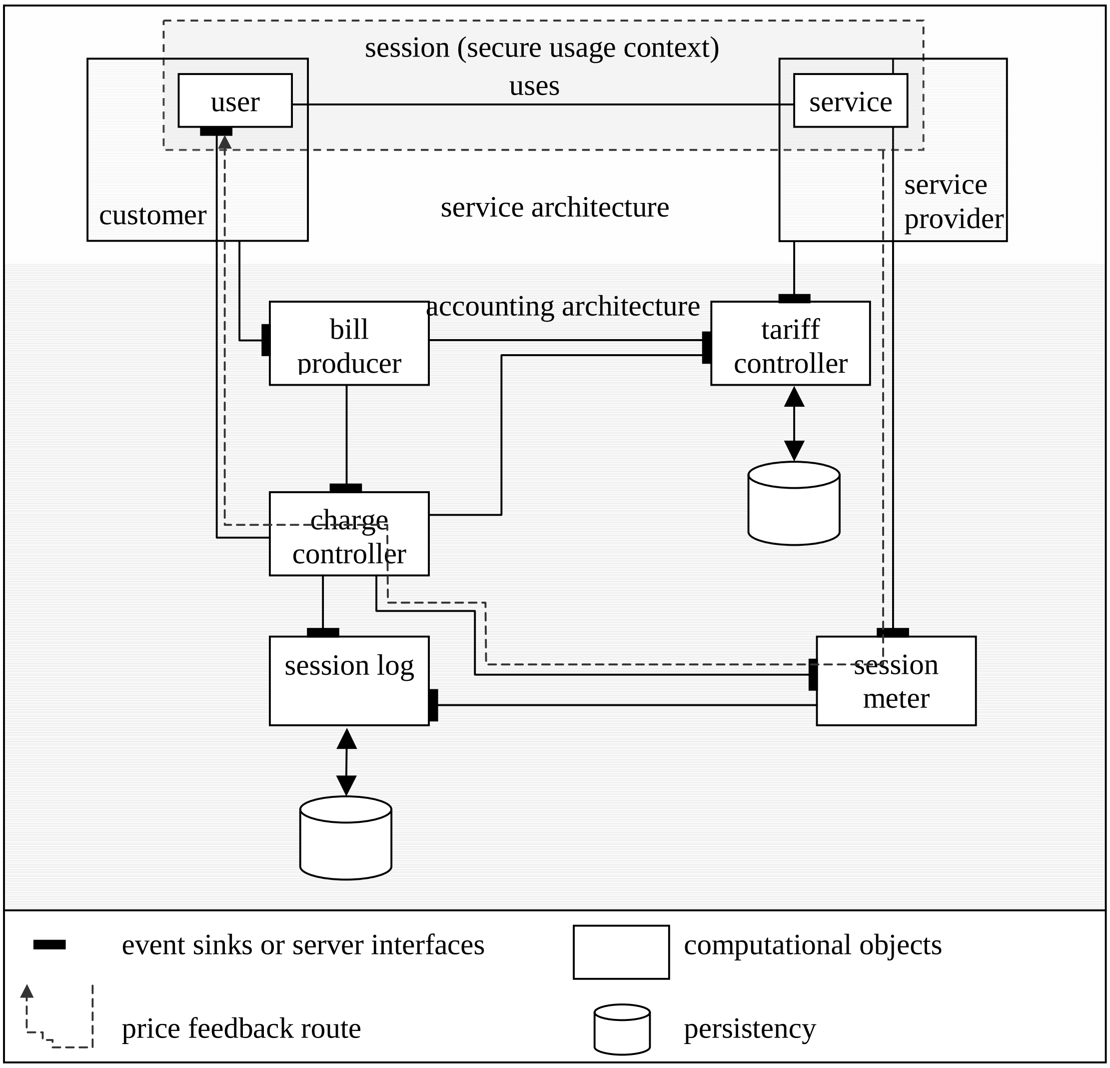


Figure 4.2, Overview computational model

This diagram shows the overall division of functionality between the sub-components of the architecture, the interactions between them, their general domains, and some of the supporting concepts imported from the TINA service architecture.

The top of the figure shows the components and concepts of the service architecture that are necessary for the accounting architecture. The user and service objects server generalise the computational objects related to these roles in the service architecture for the sake of clarity; details of the computational objects involved are in section 3.3.3.3, and details of instantiations can be seen in sections 5.2.2.1-5.2.2.4 and 5.3.2. The user and service objects belong to the customer and service domains, respectively. These domains correspond to the roles of the same name in the TINA business model discussed in section 4.2.

When a new service is created, it must register its tariffs with the tariff controller, which then stores them. Details of the tariff information model are left to the next section, but while different tariff objects correspond to subscribers and users, they are no longer seen as sub-types of the same object as they were in the TINA tariff model shown in 3.4.5.1; in particular the user specific tariff contains information necessary to meet price adaptivity requirements.

As a user uses it, the service emits accounting events corresponding to entries in one of its tariffs, which can be registered on a per-user, per service basis. These are collected by a session meter which is created by a session meter factory (not shown for clarity) as part of the service session start-up. A user’s accounting events, associated with the service session, form the main link between the service session and a separate ‘accounting session’ which exists between the service components and the session meter. As well as the inherent service session association, the events are also related based on the party producing them and the user represented by that party, where a party represents an instance of user participation in the service session. The charge controller can produce charges based on these events, their relatedness, and the pricing information contained in a tariff, then display the cost of service usage on a per-session or per-user basis in real time. The overall price feedback route is shown by the dashed line.

The bill producer, charge controller and session log together act to provide most common form of service charge production; that of a time delimited bill. The request for a bill may happen on a periodic basis decided as part of the overall contract between the customer and service provider; this aspect is outside the scope of the accounting model. The bill is produced based on session usage data persisted in the session log, and, depending on the time period, data from on-going sessions in the session meter. The charges which make up the bill are produced, based on the relevant user tariffs, by the charge control. These are then used to produce a bill by the bill producer, which also adds service subscription charges. The information model for charging and billing are also detailed in the next section.

4.5.2 Information model

Figure 4.3 shows an overview information model for the contract and tariff elements of the architecture. This diagram uses UML notation to show the relationships between the information objects. The lighter shaded region in the top region of the diagram shows relevant parts of the subscription information model.

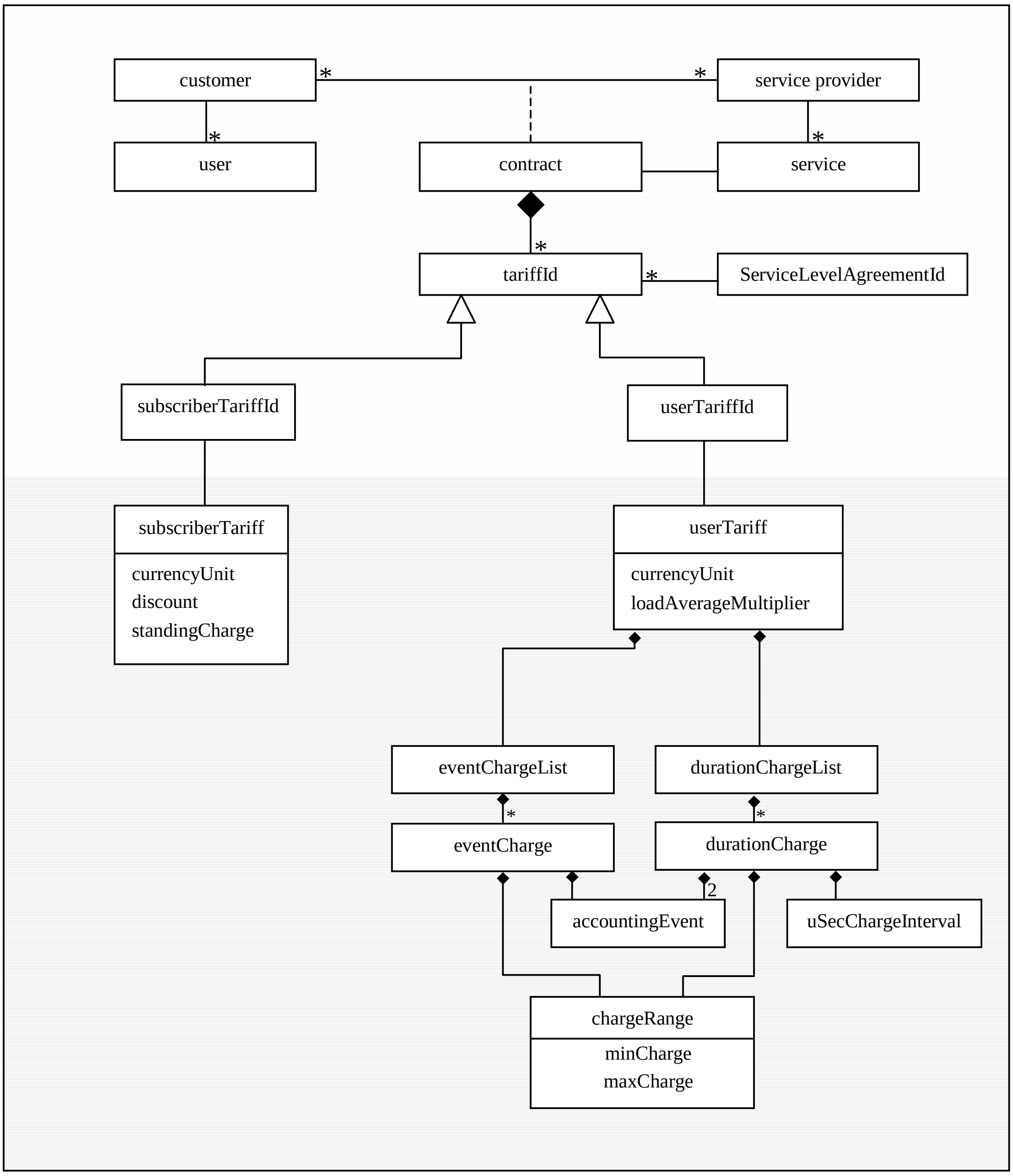


Figure 4.3, Overview tariff information model

The subscription information model shown represents the association between a customer and a service provider as a contract, which may be for the supply of more than one service. The contract contains a list of tariff identifiers corresponding the users for each service and an identifier for an overall service subscription tariff. Each tariff identifier can also be associated with an overall 'level' of service, but this is seen as a service subscription issue. The accounting model only requires that the service customer supply tariff identifiers when requesting real time charges or a bill; as such, it does not insist on the subscription model shown.

The periodic subscription charge, usually levied as part of a bill, is described by the '`subscriberTariff`', which lists a standing charge and a discretionary discount. This differs little from the basic tariff described in the TINA accounting model. The '`userTariff`' consists of two lists, one of which contains event based charges, with a corresponding price range and the other of which contains two events which delimit a time period, measured in units of the '`uSecChargeInterval`', also with a corresponding price range. The tariff also specifies a '`loadAverageMultiplier`' which dictates how sensitive prices should be to service load indications. (An implementation of a pricing mechanism based on this model is described and evaluated in section 6.3).

Figure 4.4 shows parts of the information model relevant to charging and billing.

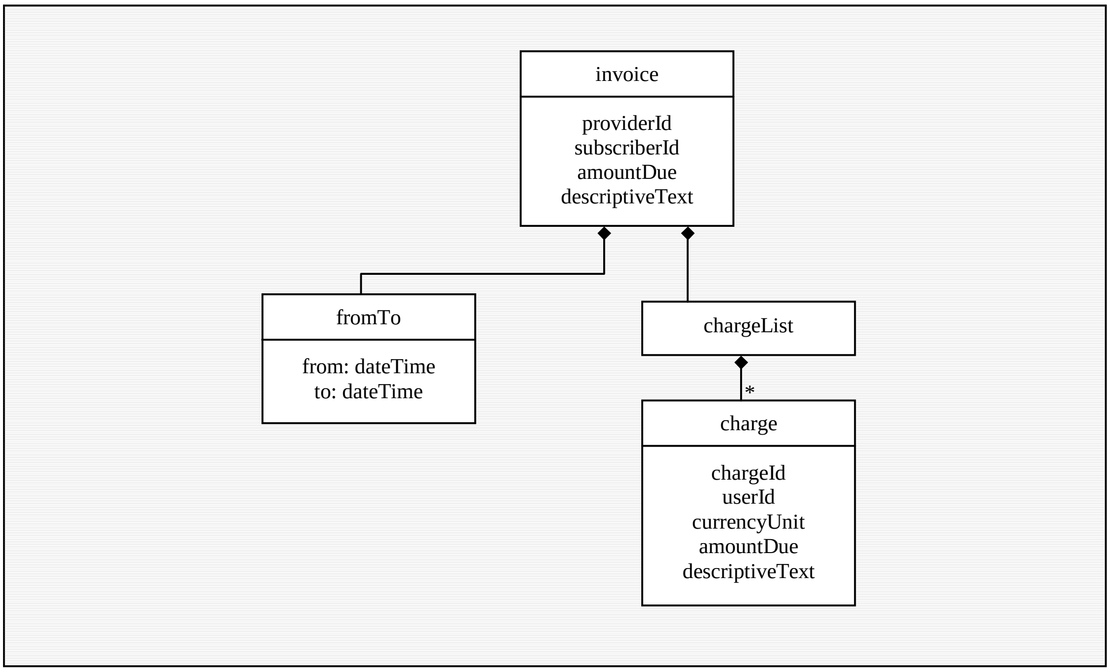


Figure 4.4, Overview billing and charging information model

This model shows a periodic bill, covering a period delimited by the '`fromTo`' structure, consisting of a list of charges, grouped on a per user basis. The charges can be calculated by the charge control according to the user specific tariffs and based on the user's service usage information within the sessions recorded by the session meter log and/or available from the session meter itself.

4.6 Summary

This chapter initially outlined the influences and attendant requirements for a generic service accounting architecture. It suggested that such an architecture should facilitate the speedy creation of new, accountable, services, helping to reduce their time-to-market. It was argued that, to do this, the architecture must consider, and take advantage of, existing DPE based service architectures that themselves support rapid service development through service component re-use. To facilitate accounting component re-use, network independent, service based charging was mooted, along with demand based pricing mechanisms that can enable real-time cost feedback to service subscribers.

The architectural principles that flowed from these market influences and technical requirements were detailed in section 4.4, they included; a session based accounting context facilitating service based accounting, extensible tariffs as part of a service description for accounting purposes, a separation between network based services and accounting for those services to support component based re-use, tariffs capable of being capped and real time cost feedback mechanisms. The assessment of whether a service accounting system built according to these principles meets the requirements imposed is presented in chapter 6, in the context of an overall architectural evaluation. Section 4.5 presented overview computational and information models of the architecture.

The next chapter details the design and iterative implementation of sets of accounting components which were used to realise the architecture.

5 Design and implementation of accounting components

5.1 Introduction

The previous chapter described an architecture embodying a set of concepts necessary to account, price and charge for network based services. At the implementation stage, this research was concerned with the development of service level accounting components according to the architectural principles specified therein. In the course of this research a sample Web based information service and service management components were also developed to help illustrate service level accounting. This chapter describes implementations that demonstrate an incremental application of the service accounting architecture, over the course of two European ACTS projects, to four separate services, covering two different service paradigms.

The first of the ACTS projects was Prospect [PD22B98]. Prospect concerned itself with the management of integrated services. It chose a business model in which one service provider could integrate multiple services to offer an integrated service to its customers. This service provider offered a tele-educational service (TES), composed of three separate services and supported by a connectivity and basic network service. Each service was provided to the TES by a separate, physically remote, organisation. One such constituent service was the Hyper Media service (HM), which was representative of information, or on-line services, while another, the Multimedia Conferencing service (MMC) was representative of a more traditional real time communication based service, albeit over a non-traditional transport network. These services are the epitome of different service paradigms available to-day; the information service was a non-interactive, relatively low bandwidth application, while the video-conferencing application was interactive and bandwidth intensive; nevertheless, common accounting components were developed for both the HM and MMC services as part of this research.

The second ACTS project was the Flowthru project [FTD800]. This project set about reusing components from other projects (including Prospect) to show how different integration methodologies and technologies could be applied to create reusable telecommunications management components. The accounting components developed as part of this research were used to account and charge for the usage of a digital library service (‘DigLib’) and a desktop audio-video conference service (‘DVAC’) at the service level, and aggregate network based charges to produce a bill. Both of these services originated within the ACTS VITAL project [VitalA298]. Extensions to the system previously developed during Prospect allowed real time charging systems with desktop displays (‘Hot-billing’) and full run-time re-use of the accounting components by different services. The components were used within two different integration frameworks; the CORBA based TINA framework and that of a CORBA based workflow system.

5.2 The Prospect project

This section describes an implementation, as part of this research, of a service based accounting mechanism in the context of the Prospect project. First it describes the background to the project, presenting an overview of the implementation work done towards the Prospect trials, describes these trials, then evaluates the work carried out, pointing out limitations in the approach followed.

5.2.1 Background

The Prospect project originated from an earlier ACTS project, Prepare [Prepare96], which had implemented TMN based (see section 3.3.2) multi-domain management systems. Prepare’s approach was based on administrative domains which were intended to enable service provider autonomy and encourage co-operation between providers; this differed from other approaches where domains were technology based and served only to partition management responsibility within a provider. Prepare introduced the concept of the ‘Open Services Market’, recognising the growing importance of software based services in the light of the European Commission’s telecommunications service and ‘Open Network Provision’ directives [90/388/EEC], [90/387/EEC]. Prepare itself concentrated on bearer services (e.g. Virtual Private Networks, VPNs).

Prospect chose a business model which it expected to exist in the ‘Open Services Market’ identified by Prepare, selecting business roles similar to those used in the TINA Business Model [TINABM97], i.e. based on a customer/provider dichotomy, as a basis for its administrative domains. The generic business model is shown in Figure 5.1. In this model there are two primary stakeholders; the service Customer and the service Provider, each supporting several roles. The service provider can be further specialised to be: a tele-service provider who provides basic service functionality, e.g. a video conferencing system; a composite service provider, who composes end-user services from the tele-services; a network operator, who provides basic network connectivity; and a communication service provider, who provides a value added service over basic network connectivity, e.g. a Virtual Private Network provider.

The roles supported correspond in most cases to those in the TINA Business Model. The customer can play the roles of a service end-user or a customer administrator, who manages subscriptions for other users and can request a bill for service usage. The provider supports the role of a provider administrator, who manages subscription information and communications service support for the constituent tele-services. Prospect specifically made the role of a system integrator separate; in this case it differs from the TINA Business Model. The system integrator role is necessary to show the value added by the composite service provider in integrating the tele-service offerings to provide a composite service.

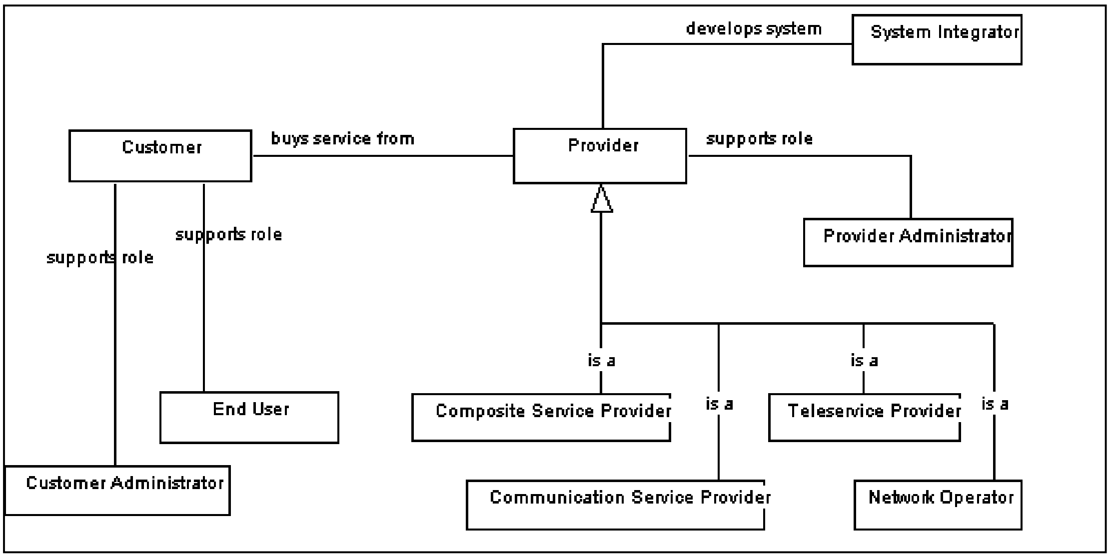


Figure 5.1, The Prospect generic business model [PD14b98]

An instantiation of the generic business model, chosen as the basis for the majority of the Prospect trials, is shown in Figure 5.2. In this model a reseller could integrate multiple services (‘Multimedia Tele-Services’ or MMTSes) to offer its customers an integrated service. Tele-education was chosen as a representative service, with the service provider known as the Tele-educational Service, or TES, provider. The TES provider constructed its service using these three MMTSes:

- an Hyper-Media (HM) service, developed as sample service to aid in this research work; this offered secure accounted access to the facilities afforded by a web server,
- a Multimedia Conferencing (MMC) service, which offered an Internet based multicast (M-Bone) video conferencing service, and
- a Web-Store (WS) service, offering web based storage facilities.

The figure shows an end-user service, the Tele-Educational Service (TES), composed of three separate tele-services; an HM service, an MMC service and a WS service, also shown are the TES customer roles (customer administrator and end-user) and the TES provider administrator role. The system integrator role is fulfilled in the pre-service phase; the trial model shows only the in-service phase. The customer administrator is a user of the TES customer management service, composed from the constituent tele-services' management services. The figure also shows a VPN provider and ATM provider as communication service provider and network operator respectively.

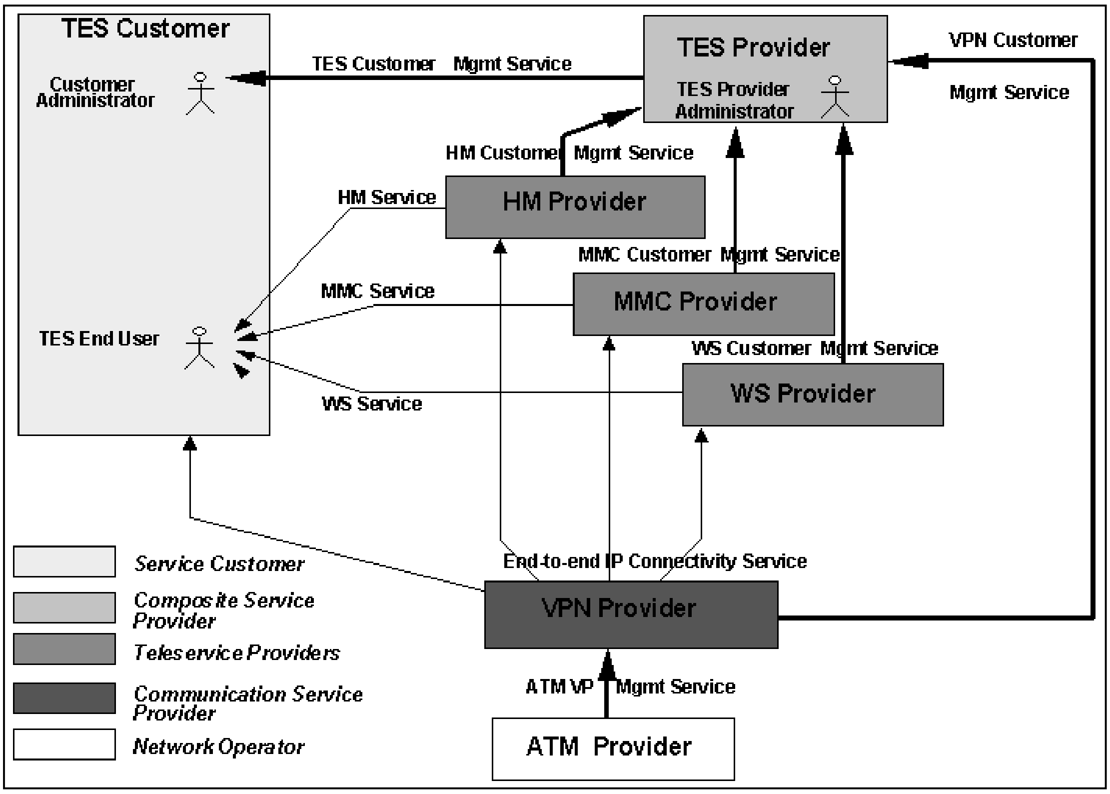


Figure 5.2, The chosen Prospect business model [PD14b98]

Prospect produced federated service management systems, composed from several service providers' management systems, based on this business model. Each management system was composed from components, including reusable service components, following the adopted TINA service architecture (see section 3.3.3.3). Although seeking to adopt a TINA based approach to service management, at the time of project inception many of the TINA standards were as yet unpublished, or at least not generally available. This necessitated an approach that used published TINA architectural concepts, but could not directly make use of unpublished TINA information model definitions.

5.2.2 Project implementation overview

In the absence a TINA Distributed Processing Environment (DPE) implementation, and any suitable DPE with the stream oriented interfaces required, Prospect took a pragmatic approach. It adopted the CORBA architecture as a basis for the creation of services that did not need the specialised communication facilities offered by the stream interfaces, i.e. Internet protocol (IP) based services.

Three ('Multimedia Tele-Services' or MMTSes), the Hyper-Media (HM) service, the Multimedia Conferencing (MMC) service and the Web-Store (WS) service were integrated to offer an integrated Tele-educational Service (TES), provided by a separate TES provider. In this scenario network connectivity was provided as just another service, with charges for it consolidated in an overall bill by the TES provider to facilitate 'one stop shopping'. Tele-services could only charge for the added value they gave over network connectivity; this led naturally to service based accounting in a way analogous to thinking about services in an era of free bandwidth.

The TINA service architecture included facilities for session co-ordination at the service level, encapsulated in a Service Session Manager (SSM). This component's main purpose was to 'keep track [of] and control the resources shared by multiple users in a service session' but specifically noted that 'compound' service offerings (like Prospect's) were for further study [TSA95]. The SSM was thus specifically orientated towards network resource allocation to support the service session; it facilitated a mapping between a logical connection graph at the service level to a physical connection graph (representing the actual network elements needed to support communication) at the network level. The actual mapping role is carried out by the Communication Session Manager (CSM) in the TINA service architecture.

In the absence of specific communication session management, Prospect rendered an explicit Service Session Manager unnecessary and introduced the concept of an Integrated Session Manager (ISM) to mediate service independent interactions between the constituent services. This supported hierarchical session relationships, allowing the creation of what appeared to be a single service session from the constituent service sessions. Prospect simplified the service session it needed to support by removing the requirements for group session functionality and allowing this functionality to be provided by service specific components (e.g. in the video conferencing service). This was later to affect the re-use of the service accounting components, first developed as part in of this research within the context of Prospect, in true TINA environments (see section 5.2.5).

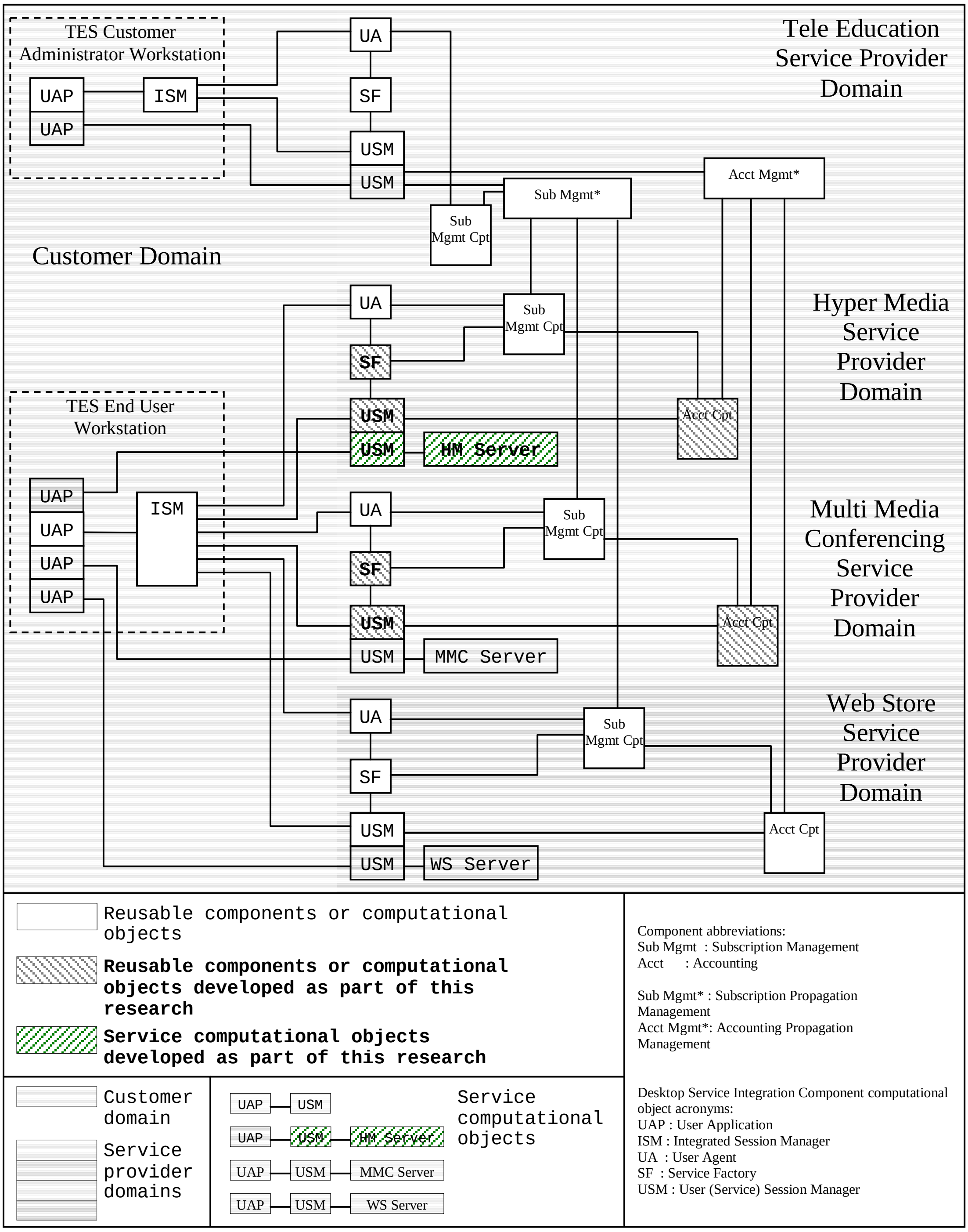
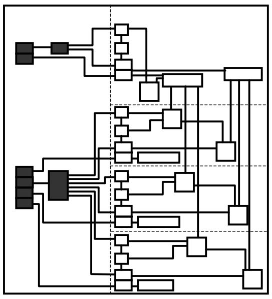


Figure 5.3, An overview computational model of the service control and management systems in Prospect, based on that in [Lewis97b]

Figure 5.3 outlines the components and the component interactions that comprised the majority of the Prospect trial systems. It shows an end-user using the services of three separate service providers, oblivious to their existence as separate entities as they transparently contribute to an overall tele-educational service composed and provided by the tele-educational service (TES) provider. It also shows a customer administrator using the management services of the TES provider which inform and rely on the management services of the constituent services.

The following sections highlight and explain portions of this overall model, detailing the interactions between the computational objects and components. The components and computational objects developed as part of the present research are shown in bold with diagonal hatching; the computational objects comprising the accounting component (‘Acct Mgmt’), developed as part of this research, and their interactions are specified in more detail in section 5.2.3.

5.2.2.1 The User Application and the Integrated Session Manager

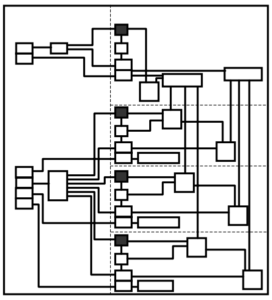


The UAP -sometimes abbreviated ‘UAp’- is the TINA term for the user application which an end user can use to interact with any service offering. Prospect was committed to using ‘off the shelf’ software where possible to avoid investing effort in what was deemed an inessential part of the problem being researched. In effect, this meant that web browser software, specifically a ‘Netscape’ browser, was used to provide a unified user interface to all the service offerings, e.g. when a user entered a virtual lecture room in the tele-educational course, video conferencing software would start automatically. Different service applications were loaded into the browser in the form of Java applets that interacted and co-ordinated in a manner hidden from the end user.

The Integrated Session Manager (ISM) co-ordinated the MMTS service sessions comprising the TES service session. There was an instantiation of the ISM for each node participating in a trial; it did not have to run on the same node as the UAP, but usually did. The ISM composed an hierarchical service session from the MMTS service sessions, providing a mapping between service session identifiers; this meant that when a TES service session was terminated by the end user, all the MMTS service sessions that constituted it were also terminated.

The Prospect ISM fulfilled the functions of the TINA SSM with regard to session control; it created an access session which allowed an authenticated user to obtain information on which services they were authorised to use. The user could then choose to access a specific service, in which case the ISM facilitated the management interactions between the customer and service domains by invoking service control operations on the relevant service UA, user agent, and USM, user (service) session manager (see following sections). The service specific user applications are highlighted in the customer domain, shading showing the affiliation in each case. These UAPs communicated with service dependent USMs, which in turn interacted with the actual service implementation (shown by the servers in each case) to effect service functionality.

5.2.2.2 The User Agent



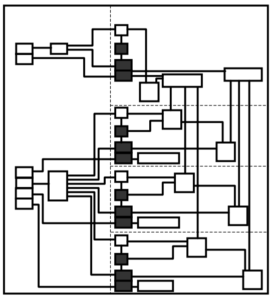
The UA, or user agent, represented a user in a service provider domain. In the TINA service architecture it is a necessary mediator in service session creation because a service provider can have more than one service; in Prospect it mediated between the user and the Subscription Management component of the provider's system, which specified the services the user could access. The UA and the ISM were participants in the access session (see section 3.3.3.3). Figure 5.4 shows part of the IDL definition of the access control interface of the user agent used in Prospect.

End user authentication information, entered via the service independent UAP (shown in white in Figure 5.3), was used by the ISM in an invocation of the '`createAccessSession`' operation on the user agent in the relevant service domain. Other information included a terminal identifier (e.g. a workstation IP address) and a network access point (e.g. a router IP address for that network). If the invocation was successful, an access session was created for that user; the access session identifier was returned with a list of available services. The ISM could then mediate the selection of a service offering from each provider; a service was chosen from the list of available services, then its identifier and the access session identifier were passed as parameters to the '`selectService`' operation. The UA then interacted with the relevant service factory (SF) to create a service session, the service factory returning a service session identifier and a service independent USM reference (see next section). The ISM could then effect service session control through this reference.

```
////////////////////////////////////////////////////////////////////////////////
// UA Access Control Interface (i_uaAci)
//              this interface allows the client to request
//              access to services. Methods include authentication,
//              getting a list of subscribed services, select a service.

interface i_uaAci {

  void createAccessSession (
                in              t_UserId user_id,
                in              t_Password password,    // Can be a signature.
                in              t_TermId term_id,
                in              t_NapId nap_id,
                out             t_AccessSessionId access_session_id,
                out             t_AvailableSvcList svc_list
) raises (
                e_dsiAccessDenied );

  void deleteAccessSession ...

  void selectService (
                in              t_AccessSessionId access_session_id,
                in              t_SvcSelection svc_selection,
                out             t_SvcSessionId svc_session_id,
                out             t_IntRef usm_ref,
                out             t_Password svc_session_key
) raises (
                e_dsiAuthorisationFailed,
                e_dsiInvalidSvcSelection );

  void unselectService ...

}; // end of i_uaAci interface
```

Figure 5.4, Part of the access control interface definition of the UA [UA.idl]

5.2.2.3 The Service Factory and the User Session Manager



The service factory (SF) enabled the creation of the service sessions that allowed end-users to use a service. The service factory created user (service) session managers (USMs) that were subsequently used to control the service session. Figure 5.5 shows the IDL of the service factory session lifecycle control interface used in Prospect.

A service session creation request began when the end user selected a service through the user agent; to fulfil this request, the user agent invoked the ‘`createUSM`’ operation on the correct service factory. The parameters included a user and account identifier, a password, and service identification information. If this operation was successful, a service session was created and its identifier and a service independent USM reference were passed back to the ISM via the user agent.

```
////////////////////////////////////////////////////////////////////////
// Interfaces for the Service Factory CO

////////////////////////////////////////////////////////////////////////
// SF Session Life-cycle Control Interface (i_sfSlci)
//              this interface allows the client to request
//              control over the life-cycle of a user’s USM CO
//              It is used by the user agent when it wants to create a USM
//              for a user

interface i_sfSlci {

  void createUSM (
                in              t_UserId user_id,
                in              t_SvcId svc_id,
                in              t_SvcProfile svc_profile,
                in              string account_no,
                in              t_Password svc_session_key,
                out             t_SvcSessionId svc_session_id,
                out             t_IntRef usm_ref)
  raises (e_dsiInvalidSvcProfile);

  void deleteUSM (
                in              t_SvcSessionId svc_session_id);

}; // end of i_sfSlci interface
//
////////////////////////////////////////////////////////////////////////
```

Figure 5.5, Service Factory interface definition [M_sf.idl]

The USM was defined in two parts; a service independent part that covered service session control, including creating and terminating a service session, and a service dependent part that gave the end user access to service functionality. The ISM interacted with the service independent parts of the USM to effect session control, and the UAP interacted with the service dependent parts to use the service.

Figure 5.6 shows the definition of the interfaces that constituted the service independent parts of the USM computational object. The lifecycle control interface (‘`i_usmLci`’) was used by the service factory to create the actual USM objects; each object implemented an instance of the session control (‘`i_usmSci`’) interface, i.e. one for each service session. A reference to the service independent USM object was returned to enable service session control. When a service session was started by invoking ‘initiate’ on the session control interface, information on how to contact the service was returned to the to the user. This contact information consisted of a URL of a specific MIME type containing a reference to the service dependent USM which that service’s UAP applets could recognise and act upon if necessary.

```
module USM {

//////////////////////////////////////////////////////////////////////
// Interfaces for the User Session Manager CO

//////////////////////////////////////////////////////////////////////
// USM Life-cycle Control Interface (i_usmLci)

interface i_usmLci {

  void initialise (
            in          t_UserId                  user_id,
            in          t_SvcSessionId            svc_session_id,
            in          t_SvcProfile   svc_profile,
            in          t_IntRefList   input_int_ref_list,
            in          string                    account_no,
            in          t_Password     svc_session_key,
            out         t_IntRefList   generated_ref_list );

  void remove (in t_SvcSessionId  svc_session_id);

}; // end of i_usmLci interface


//////////////////////////////////////////////////////////////////////
// USM Session Control Interface (i_usmSci)
//          this interface allows the client to request
//          control over service sessions. Methods include initiate,
//          terminate a service session on the user’s side

interface i_usmSci {

  void initiate(
            out         t_SvcRefList svc_specific_ref_list);

  void terminate();


}; // end of i_usmSci interface

#include "../service_specific/i_usmSdi.idl"

}; // end of module USM
```

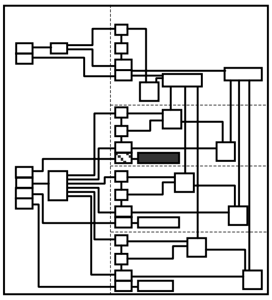
Figure 5.6, USM interface definition [M_usm.idl]

As the user used the service, the UAP caused invocations to the service dependent interface of the USM object. This interface is included in the module definition as ‘`i_usmSdi.idl`’ above. This interface of the USM object was service dependent and was defined and implemented separately for each service; the next section describes the service dependent interface for the HM service.

The USM was responsible for the generation of accounting events in response to both service session control and actual service usage. These events were collected by that service’s instantiation of the accounting management component and used as a basis for accounting.

The service factories and the USMs for the HM and MMC services were developed as part of this research.

5.2.2.4 The Hyper Media Service



The HM, or Hyper Media, service was developed as part of this research to enable secure, accounted HTTP (hypertext transfer protocol) access to the courseware resources developed as part of the project. It also served as a sample service to test the functionality of the computational objects developed as part of this research.

Initially the HM service consisted of a modified ‘Apache’ web server with a firewall based access control mechanism [Lewis97]. The earliest modifications to the server allowed ‘Cookies’ (an application level state mechanism then only beginning to be employed in HTTP) to be used as a basis for accounting [RFC2109]. The modifications consisted of a lightweight CORBA client of the USM service dependent interface built into the user tracking module of the web server; this enabled the tracking of service usage within service sessions. The USM service dependent interface is shown in Figure 5.7

```
interface i_usmSdi {

// deprecated  in trial 2.x
//void create_proxy(in string userid, out long port) raises     (e_usm_sdi_UserNotRegistered,e_usm_sdi_NoMoreUsers,e_usm_sdi_Another);
//void kill_proxy(in string cookie) raises (e_usm_sdi_ProxyDoesnotExist);

// user tracking part
  oneway void usage(in string service_session_id, in string URL, in short status);
};
```

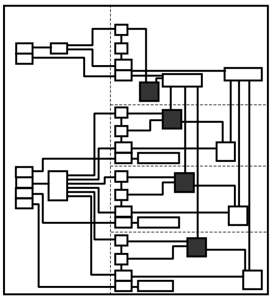
Figure 5.7, USM service dependent interface definition for the HM service

Access control was effected by the use of a firewall mechanism. The modified server was either kept behind a firewall, or configured only to accept requests from its own domain. When a service session was started, the service dependent service reference returned by the ‘`initiate`’ operation of the USM session control interface consisted of the URL of a CGI script on an intermediary server running on a firewall machine. When the UAP accessed the URL, the script executed, creating an application-level HTTP proxy on the firewall machine then redirecting the browser to the service proper through this proxy. The proxy thus enabled access to the modified server and provided a means of graceful service shutdown and barring; unfortunately, the proxy also meant that courseware pages could not be book-marked in the browser as the URL of the proxy was liable to change between service session instances; this was circumvented by the use of a service side bookmark scheme that also enabled notes (this was developed as part of the courseware).

The second version of the HM service used a modified access control scheme developed in response to the complications and relative unreliability of the original scheme. This took advantage of the fact that the ‘Apache’ API allowed the developer to easily modify the behaviour of the server for certain MIME-types, i.e. it enabled the building of a service specific MIME-type handler that executed based on the file name extension of the URL file [Redmond97c]. Now when a service session was started, the service dependent service reference returned by the '`initiate`' operation of the USM session control interface consisted of the URL that contained an encoded version of the service session identifier. The MIME handler in the server recognised this and decoded the service session identifier to use as a basis for the 'cookie'. Access control was effected because other modifications ensured that service access (excepting the service specific MIME-type) was only possible with an appropriate 'cookie'.

Both versions of the HM service reported service usage within the service session via the USM service dependent interface; this then generated the appropriate accounting events used as a basis for service accounting.

5.2.2.5 The subscription management component

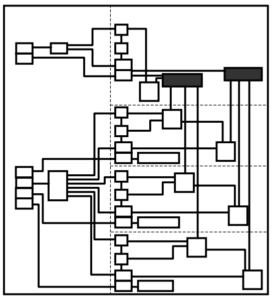


The subscription management component allowed for the authorisation and barring of users' access to specific services and the addition and removal of network sites that supported these users. This component was re-used by the HM and MMC MMTSes and the TES service, which forwarded its subscriber information to the constituent MMTS services via the subscription propagation component (see next section).

The subscription management component was queried by the UA as part of service session set-up to ascertain whether a user was entitled to access a particular service. The service template handler (STH) computational object of this component was also used by service factory computational objects to enable user agents to access them; they published the session lifecycle control interface reference in the service template upon initialisation.

The accounting component interacted with the subscription registrar computational object of subscription component to find the identifiers of the tariffs the service customers had agreed to as part of their contract.

5.2.2.6 The propagation components



The propagation components (‘Sub Mgmt *’ & ‘Acct Mgmt *’) were used only within the TES service provider domain to facilitate the Prospect design goal of ‘one-stop-shopping’.

The subscription propagation component ensured that each constituent MMTS of the TES service received the same customer subscription information and updates to that that information. The accounting propagation component was used to request a bill from each MMTS service used to constitute the TES when the TES customer requested a bill; it then combined the bills to produce a single bill.

5.2.3 Accounting component implementation

The computational objects comprising the Prospect accounting component and their interactions are shown in Figure 5.8 below.

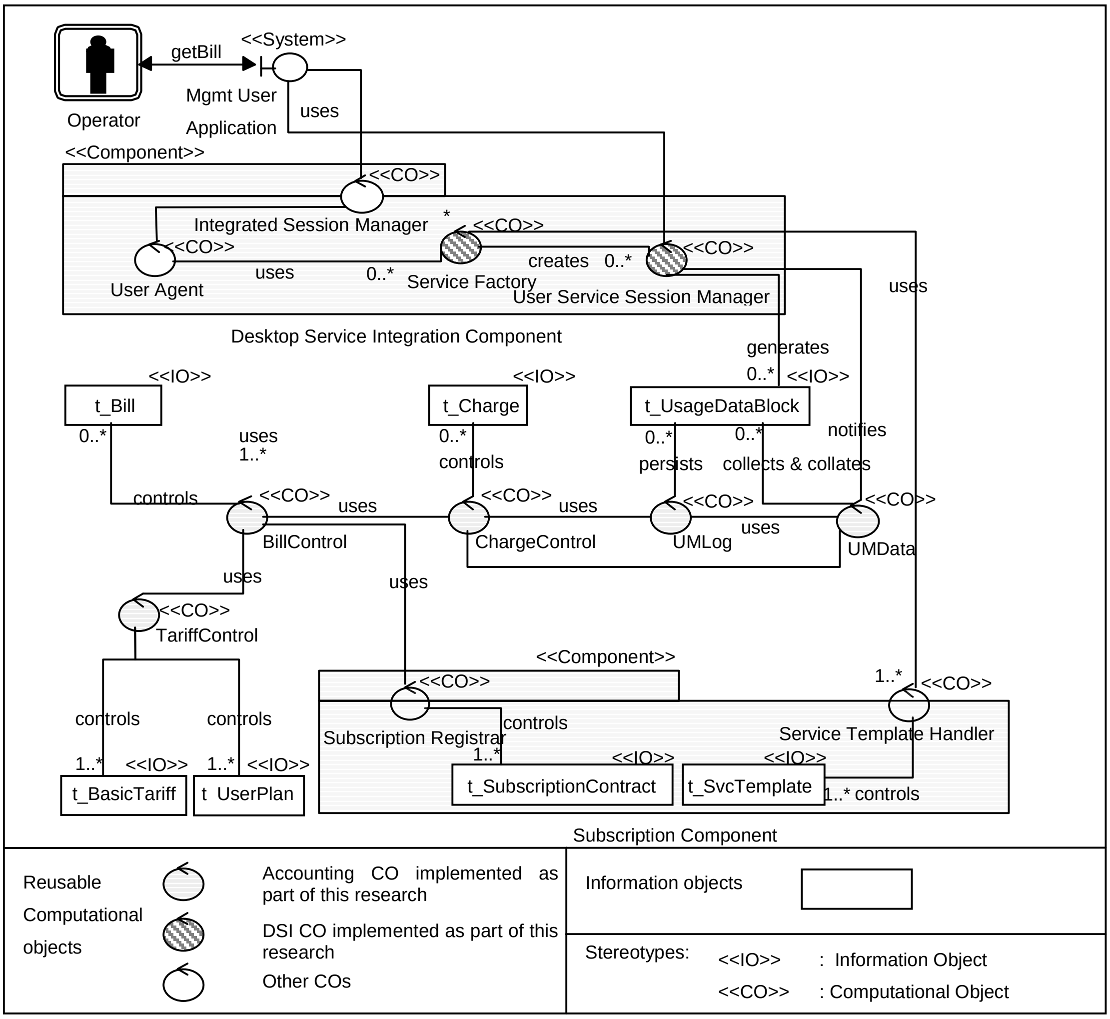


Figure 5.8, Prospect accounting component objects and their associations with other components.

This figure uses the UML package construct with the computational objects differentiated between entity, control and interface objects following the notation defined by Jacobson in [Jacobson92] in the absence of a mature UML notation at the time of implementation design and following project notation guidelines. It shows the interactions between the computational objects that constitute the Accounting, Desktop Integration & Session Control, and Subscription components of the Prospect Project. Computational objects developed as part of this research are shown either shaded or with diagonal hatching.

The figure shows the Customer administrator user as the ultimate end user of the accounting component functionality, requesting a bill via the Management Application. While this is the case in Prospect, several sets of interactions must already have taken place to give a basis for producing the bill. As detailed earlier, the interactions started when the service factory created a new service session upon request from the user agent (UA); it generated a new service session identifier and requested a new user (service) session manager (USM) object via the ‘lifecycle control’ interface of the USM computational object.

Each USM object implemented two CORBA interfaces: the Session control interface (Sci) and the Service dependent interface (Sdi). The session control interface gave general service session control functionality to each user (Prospect service sessions only had one user, see section 5.2.5). However, each USM object also had a corresponding usage data meter in the Accounting Component’s Usage Meter (UMData) computational object. As control operations (initiate, terminate) were invoked on the session control interface via the Integrated Session Manager (ISM), service independent accounting events were generated by the USM - in the absence of a TINA Service Session Manager (SSM) - and passed to the appropriate usage meter data object. Service dependent operations on the service dependent interface produced similar service dependent accounting events.

The events thus generated by service users formed the basis for computing the bill when it was requested by a customer administrator (where a customer, or subscriber, could have many users). The interactions that occurred when a bill was requested began when the management application invoked on a separate User Agent (UA) requesting a bill for that subscriber’s service usage. This request was passed onto the BillControl computational object of the accounting component, which then requested the Service Contract from the Subscription Registrar (SubR) computational object of the subscription component. This contained tariff identifiers that the BillControl used to fetch tariffs from the TariffControl computational object. The tariffs, a basic tariff and a user plan, corresponded to service usage based charges and subscription based charges.

The BillControl requested charges for service usage from the ChargeControl computational object, sending the appropriate tariff with the request. The ChargeControl then requested the service accounting events that occurred in present or previous sessions from the UMData and the UMLog computational objects, where the UMLog computational object was used to persist session usage data by the UMData computational object. The charges were then calculated, based on the tariffs, for service usage within the service sessions and passed back to the BillControl, which could then apply a subscriber specific tariff to compute a final bill, passing it back to the management application on the customer administrator’s desktop.

The next sections detail the implementation of the computational objects used to produce the bill for the customer administrator in the scenario given.

5.2.3.1 The usage meter

The usage metering data, or UMData, computational object built for the first Prospect trial was based on an outline information and computational model, part of the TINA service architecture, made available at the time of project inception; detailed specifications were as yet unavailable [PD4B98]. The model bore similarities to the X.742 'Usage Metering Function for Accounting Purposes' [X.74298], where events generated by accountable resources were transformed into usage metering records as part of the usage metering process and then used to calculate charges based only on recorded resource utilisation. Subsequent document availability has borne this out [TINAAA96]. This research thus inherited an initial accounting system specification based on that defined in X.742; in some cases a transliteration of the ASN.1 therein specified.

The UMData gathered the accounting events generated by the USM. The initial specification mandated one UMData Computational Object (equivalent to a CORBA server) per USM, but a less resource intensive implementation was actually built; the initialisation interface was used as a factory interface producing lightweight CORBA objects corresponding to each USM (see Figure 5.9), these implemented two interfaces (management and query) per object (see Figure 5.10).

```
module m_UMData {

   interface i_UMDataInit {
              exception e_accNoInstantiation { string reason; };
              exception e_accCouldNotInstantiate { string reason; };

              void init ( in string account_no, in t_SessionId sessionId,
                             out t_IntRefList generated_ref_list,out short instance )
                raises (e_accCouldNotInstantiate);

              void terminate (in short instance) raises (e_accNoInstantiation);

              void terminateAll() raises (e_accNoInstantiation);

              void getQueryRefs (out t_IntRefList generated_ref_list)
                raises (e_accNoInstantiation);

   };

    ...

 };
```

Figure 5.9, Trial 2.1 UMData computational object definition showing the factory interface

```
 module m_UMData {

    ...

    interface i_UMDataMgmt {
       exception e_accBadUsageDataBlock { string reason; };

                    void addUsageData ( in t_UsageDataBlock usageData )
                      raises (e_accBadUsageDataBlock);

   };

   interface i_UMDataQuery {

                    readonly attribute string account_no;
                    readonly attribute t_SessionId sessionId;

                    t_Account_SessionUsageData getUsageData ();
   };
};
```

Figure 5.10, Trial 2.1 UMData computational object definition showing the interfaces implemented by each UMData object

These lightweight CORBA objects were instantiated for a given customer participating in a service session. Thereafter the usage data received was collected and collated based on the subscriber-user relationship supported by the subscription component, i.e. according to the generating user. The subscription model featured a customer, represented by an account number, with authorised users, represented by user identifiers. The customer itself was represented by a subscriber identifier which was an account number; these monikers were frequently used interchangeably. The generated data was thus grouped according to the generating customer, then the generating user; this simplification mirrored, and was a consequence of, Prospect’s adoption of single user service sessions.

The initial UMData specification for Prospect trial 1 only made provision to persist the usage data generated in the service session; no provision was made to store details of the actual service session itself. This oversight was rectified and in the initial implementation; the generated usage data was persisted with respect to the customer who would ultimately be billed for their user’s participation in the service session, but other service session details (including the session identifier) were still not persisted. By the second Prospect trial, the recognition that the context of service usage, i.e. the service session, as well as the record of resources used, could provide a useful basis for bill production [Redmond97c], [Redmond98a] led to a new organisation of the UMData and the definition of session based service accounting event persistence, described in the next section.

5.2.3.2 The usage metering log

The usage metering log, or UMLog, was originally designed to persist, or log, usage data from the UMData, in a manner analogous to the similarly named computational object in the ISO model, where usage metering records could be logged using the log control function defined in X.735 [Black95]. In the initial Prospect model for trial 1, the accountable events received from the UMData were ordered temporally, and only coincidentally had session relations thanks to the simplifying condition of single user service sessions adopted by Prospect; although the user responsible for the generating the event was recorded, other contextual information that would be of benefit in relating users in a multiparty session was not.

By the second trial it was demonstrated, as part of this research [Redmond98a], that the service usage context given by the service session could have some bearing on the calculation of charges for service usage when some usage information was not directly related to the service session lifecycle. Thus a deliberate effort was made to capture as much as possible of the service session context for the generated usage data. This necessitated few changes to the actual interface definitions as initially specified, but rather to the data structures passed between the UMData and the UMLog computational objects. Figure 5.11 and Figure 5.12 show how the information being exchanged differed between the Prospect trials. With an eye to future re-use, the trial 2 information model was kept as generic as possible; at this stage multi-user sessions were catered for in the model, if not in the implementation.

```
typedef sequence<t_UsageDataTimeStampBlock> t_UsageData;
```

Figure 5.11, Specification of information persisted by UMLog in the first trial phase

```
typedef sequence <t_UsageDataTimeStampBlock> t_PerUserUsageData;

struct t_Account_PerUserUsageData {
   string              account_no;
   t_PerUserUsageData  usagedata;
};

typedef sequence <t_PerUserUsageData> t_SessionUsageData;

struct t_Account_SessionUsageData{
    string              account_no;
    t_SessionUsageData  sessionusagedata;
};
```

Figure 5.12, Specification of information persisted by UMLog in the second trial phase

The persistence of usage data was effected by calling the store operation on the `i_UMLogManagement` interface (see Figure 5.13), supplying blocks of usage data as gathered by the UMData. This information was subsequently queried via the `i_UMLogQuery` interface as the first step in producing a bill based on a customer's service usage during a time period. The queries required and the information initially stored put few requirements on an underlying persistence mechanism; in both Prospect trial implementations, file system based persistency was used; directories represented customers, each containing several files, one to record each user's service usage.

```
module m_UMLog {

        ...

        interface i_UMLogMgmt {
                exception e_accInvalidUsageData { string reason; };
                exception e_accInvalidAccountNumber{ string reason; };
                exception e_accInvalidLoggingPeriod{ string reason; };
                ...
                exception e_accUMLogBusy{ string reason; };

                void store ( in t_Account_PerUserUsageData usageData )
                        raises (e_accInvalidAccountNumber,e_accInvalidUsageData);

                void removeLogEntries(       in string account_no,
                                                     in t_DateTime from,
                                                     in t_DateTime to )
                        raises (e_accInvalidAccountNumber,e_accInvalidLoggingPeriod,e_accUMLogBusy);

        };

        interface i_UMLogQuery {
                exception e_accInvalidAccountNumber{ string reason; };
                exception e_accInvalidLoggingPeriod{ string reason; };
                ...
                exception e_accNoLogInfo{ string reason; };

                void getLogEntries (         in string account_no,
                                                     in t_DateTime from,
                                                     in t_DateTime to,
                                                     out t_Account_SessionUsageData usageDataList )
                        raises (e_accInvalidAccountNumber,e_accInvalidLoggingPeriod,e_accNoLogInfo);

        };
};
```

Figure 5.13, The management and query interfaces of the UMLog CO

5.2.3.3 The tariff control

The initial Prospect accounting model, with the TINA model, differed from the ISO model in the identification of tariffs as a contractually specifiable link between service use and the charges that could be calculated based on that use. The model, following the TINA information model, had tariffs with two sub-types; the basic-tariff listed charges that specified how the usage data generated by the UMData, and stored by the UMLog, could be used to calculate a usage based bill, and the user-plan listed charges that were levied based only on a contractual basis; i.e. standing charges and discounts.

In the initial Prospect model it was unclear which component was responsible for the tariffs at first; the Subscription Management component listed tariffs in its information model, as did the accounting component. The tariff control object was originally specified to be created as part of the subscription to a service, but it was agreed that the specification of tariffs fitted better with the remit of the service accounting component. Thus the contracts that comprised the subscription contained only identifiers to the tariffs maintained by the tariff control. The information contained therein, thus the computational object itself, could endure beyond the lifetime of a particular service subscription. This also ensured and enforced a separation between service subscription and accounting components.

The TariffCtrl computational object was specified as a repository for both the tariff types; in the absence of sub-typing in IDL, the analysis level abstraction identified in the TINA information model was maintained through the use of an IDL union construct. In this manner the TariffCtrl initially managed both tariff types through the same interfaces, shown in Figure 5.14.

```
module m_TariffCtrl {

        ...

        interface i_TariffMgmt {
                exception e_accTariffAlreadyExists {string reason; };
                exception e_accTariffDoesNotExist {string reason; };

                void storeTariff ( in t_Tariff tariff )
                        raises (e_accTariffAlreadyExists);

                void removeTariff ( in string tariffId )
                        raises (e_accTariffDoesNotExist);
        };

        interface i_TariffQuery {
                exception e_accTariffDoesNotExist {string reason; };
                exception e_accNoTariffs {string reason; };

                void getTariff ( in string tariffId, out t_Tariff tariff )
                        raises (e_accTariffDoesNotExist);

                void listTariffs ( out t_TariffList tariffList )
                        raises (e_accNoTariffs);
        };

};
```

Figure 5.14, The management and query interfaces of the tariff control CO

The initial Prospect tariff specification was service dependent, each service having a separate list of service dependent accountable events that were interpreted by that service's charge control computation object (see the next section, 5.2.3.4; section 5.2.5 shows an example tariff). The relatively simple information structures that needed to be persisted required only a basic underlying persistence mechanism; in both Prospect trial implementations, file system based persistency was used, with tariffs automatically loaded at server initialisation.

5.2.3.4 The charge control

The ChargeControl computational object was responsible for calculating a list of per user charges for a customer's users in a given time period; this was requested by the bill control computational object as part of bill calculation. To fulfil this request, the charge control required the usage data for that customer for the given time period; these were supplied by the UMLog computational object.

The ChargeControl calculated the charges by interpreting the basic-tariff structure - supplied with the request from the bill control - matching the events in usage data blocks to the events listed therein, then adding the charge listed to the user's subtotal. Each user's usage within the given time period was used in the preparation of the per user charge; session based charging was only possible due to the existence of service independent accounting events signifying the service session lifecycle (i.e. creation and deletion, as suspension and resumption was not supported by the Prospect session model), if matching pairs occurred within the specified time period. Figure 5.15 shows the operation invoked by the bill control computational object.

```
module m_ChargeControl {

        ...

        interface i_ChargeCtrlQuery {
                exception e_accInvalidAccountNumber{ string reason; };
                exception e_accInvalidChargingPeriod{ string reason; };

                void getCharges (               in string account_no,
                                                in t_BasicTariff basicTariff,
                                                in t_DateTime from,
                                                in t_DateTime to,
                                                out t_ChargeList charges )
                        raises (e_accInvalidAccountNumber, e_accInvalidChargingPeriod);
        };

};
```

Figure 5.15, The query interface of the ChargeControl computational object

As the basic-tariff structure was service dependent, so too was the ChargeControl computational object which calculated charges based on the interpretation of this structure; each service needed its own version of the ChargeControl computational object to allow charge calculation, although source code re-use was possible and carried out.

5.2.3.5 The bill control

The BillControl computational object was responsible for calculating and presenting the bill for a customer's service usage in a given time period, as per the parameters given in a request either directly from the management application or via the accounting management propagation component.

To calculate the bill the BillControl requested the contact for that subscriber from the Subscription Registrar (SubR) computational object of the subscription management component. From this it extracted the identifiers for that subscriber's tariffs as maintained by the TariffCtrl computational object, and fetched the corresponding tariffs. The BillControl then passed the basic-tariff in the request for charges from the ChargeControl computational object. When the list of charges was returned, the user-plan, consisting of a standing charge and a discount, could then be levied on the amount of the charges before the bill was passed back to the requestor.

The BillControl also supported a simple bill paying operation that removed a customer’s log entries when satisfied that the bill total had been paid. Figure 5.16 shows the management interface of the BillControl.

```
module m_BillControl {

        interface i_BillCtrlMgmt {
                exception e_accInvalidAccountNumber{ string reason; };
                exception e_accInvalidBillingPeriod{ string reason; };
                exception e_accInvalidBillId{ string reason; };
                exception e_accSubAccessDenied{ string reason; };
                exception e_accTariffDoesNotExist {string reason; };
                exception e_accIncorrectAmount{ string reason; };

                void getBill (   in string account_no,
                                        in t_DateTime from,
                                        in t_DateTime to,
                                        out t_Bill bill )
                        raises ( e_accInvalidAccountNumber, e_accInvalidBillingPeriod,
                e_accSubAccessDenied, e_accTariffDoesNotExist);

                void payBill (  in string bill_id,
                                        in long how_much )
                        raises (e_accInvalidBillId,e_accUMLogBusy,e_accIncorrectAmount);

        };
};
```

Figure 5.16, The management interface of the BillControl computational object

5.2.4 Trial

Prospect aimed to validate the service management systems it produced in end user trials. There were two trial phases, with the second trial phase split into two parts (‘a’, related to service management and ‘b’, concentrating on network level management); each of these in turn had two trial phases, thus ‘Trial 2.1a’ concentrated on customer service management of composite services and ‘Trial 2.2a’ concentrated on Personal Communication Support (PCS) to provide per-user settings for the different services.

The first trial phase started in September 1995 and ended in March 1997, aiming to evaluate and demonstrate subscription and accounting management and secure service access. The trial consisted of the use of two tele-education courses offered by a tele-education service to establish a realistic environment for testing the management services developed. The accounting component was first demonstrated in this context and succeeded in producing a bill related to session based usage of both the video conferencing (MMC) and information service (HM); that is, the bills were generated based on service independent accounting events related to session control.

The second trial phase ran from August 1997 to August 1998. The accounting component was re-used within part ‘a’ of the second trial, in both phases (1 & 2). The trial 2.1a computational model (the basis of Figure 5.3) was the principal, and most complicated, computational model elaborated by the project, precisely because its realisation offered the most challenges. It was aimed to demonstrate the management of a multimedia service composed of services offered by separate service providers, known as Multimedia tele-service (MMTS) providers. The accounting component was re-used within both the MMC and HM services, but at this stage service independent accounting was attempted. In the end, a bill was produced by each MMTS provider and the bills aggregated at the TES level (by the ‘Acct Mgmt *’ component shown) to produce a single bill for the TES customer.

The computational model adopted for trial 2.2a was a simplified version of that adopted in trial 2.1a; in this, one service provider was used to provide access to the different MMTSes for a TES end user. This was deemed sufficient to validate the PCS features integrated into the management systems. In this trail the accounting component was deployed for both the HM and MMC services, this time provided by the same service provider.

5.2.5 Evaluation

This section evaluates the use of the accounting component in the Prospect project with respect to the architectural principles outlined in the previous chapter. These included: the adoption of a session based service architecture facilitating the use of sessions as a charge basis, tariffs that were extensible without sacrificing interoperability, a clear separation between a service and the accounting mechanisms for that service, and the use of real time cost feedback mechanisms.

Prospect enabled the specification and building of service accounting components according to TINA service architecture guidelines. These guidelines -and later the specifications as they became publicly available- did not make specific mention of service level accounting mechanisms, concentrating on network level resource usage as a basis for accounting. This caused difficulty, as the IP based services used in Prospect did not have, or need, explicit communications management; consequently, service sessions did not have corresponding communications sessions to enable the tracking of network level resource usage (see section 5.2.2).

Service level accounting as first mooted in Prospect was based only on participation within a service session, not on actual service usage, as the service session’s lifecycle was thought to mirror that of the unsupported communications session (this assumption was facilitated by the adoption of single user service sessions). The initial service level accounting within Prospect was based on service session control operations that were service independent; of the four operations specified for session control on the TINA Service Session Manager (SSM) -initiate, terminate, suspend & resume- only service initiation and termination were implemented in the Integrated Session Manager (ISM) to support the Prospect session model. Thus, while service sessions formed a basis for accounting in Prospect, it was the events controlling the lifecycle of the service session that were used initially. Nevertheless, a precedent for separate, session based, service accounting had been set within the first trial.

By the second trial, it was recognised that the service session provided a suitable context for accounting user service interactions [Redmond97c], [Redmond98a], thus the use of service dependent events as a basis for service accounting was undertaken in the second trial phase. Here the Prospect accounting model suffered from its lack of extensibility and the network level bias of the accounting components specified in the TINA service architecture; it did not have sufficient flexibility to allow service level accounting based on service functionality. This lack of extensibility was apparent at both the level of service accounting event specification and in actual service based tariff specification. While the components were developed in common between the HM and MMC services, software re-use was only carried out at the level of code re-use; each service had a service specific tariff, listing event based charges for service events, Figure 5.17 shows the tariff for the MMC service. This caused difficulty in trial 2.2a where the same provider was used for the three MMTSes and only one instance of the computational objects comprising the accounting component was necessary.

```
// MMC Specific tariff structure
struct t_BasicTariff {
  unsigned long               chargePerSession;
                                // basic charge for a session
  unsigned long               chargePerRegistration;
                                // additional charge for setting
                                // up a session
  unsigned long               chargePerRequest;
                                // additional charge for requests
  unsigned long               chargePerAccept;
                                // additional charge for accepting
                                // an invitation

  unsigned long               chargePerJoin;
  unsigned long               chargePerLeave;
  unsigned long               chargePerCreate;
  unsigned long               chargePerDelete;
  unsigned long               chargePerSecond;
};
```

Figure 5.17, The basic tariff used for the MMC service in Prospect

Service accounting with adaptive pricing and real time cost feedback mechanisms were not attempted within Prospect.

Within Prospect accounting tended to concentrate on enabling the composition of charges from the many service providers to provide a ‘one stop’ billing solution, in line with the service integration focus of the overall project. Each constituent service could produce a bill containing standardised charges that could then be used as the basis of an aggregate bill.

The accounting components developed in Prospect were modified to take account of group session semantics and network usage charges in Flowthru; modifications necessitated by the adoption of a simplified session model and the lack of separately accountable network resources in Prospect. The accounting architecture is an artefact of the application of the TINA service architecture to network based services. Its application was constrained by a greater or lesser extent by each project's adoption of the TINA Service architecture; as the Prospect project adopted a simplified version of the TINA Service Architecture these modifications were expected.

5.3 The Flowthru project

This section describes a service based accounting mechanism developed as part of this research in the context of the Flowthru project. It starts by giving some background to the project, presents an overview of the project implementation work, then describes the implementation of the accounting component and its demonstration.

5.3.1 Background

The Flowthru project arose from the combined interests of partners participating in several antecedent projects including Prospect. Flowthru aimed to re-use components from the earlier projects to show how different integration methodologies and technologies could be applied to create reusable telecommunications management components. It proposed to demonstrate solutions to telecommunication business problems using combinations of these management components. Partners participating in Flowthru included former participants in VITAL and ReTINA, both TINA validation projects. These projects were in a position to benefit from and influence contemporary TINA standards and architectures during their lifetime, but neither of them had attempted service level accounting.

Flowthru used the TeleManagement Forum's (TMF) 'Operations Map' [TMFBoM98] to identify business requirements for systems to be built using the different integration technologies and methodologies it wanted to validate. The map is shown in Figure 5.18. The map provides a reference for interactions between service providers and customers, suppliers or other service providers; the processes identified relate directly to the customer, to internal service development and maintenance and to the management of the provider's networks and systems. [FTD8].

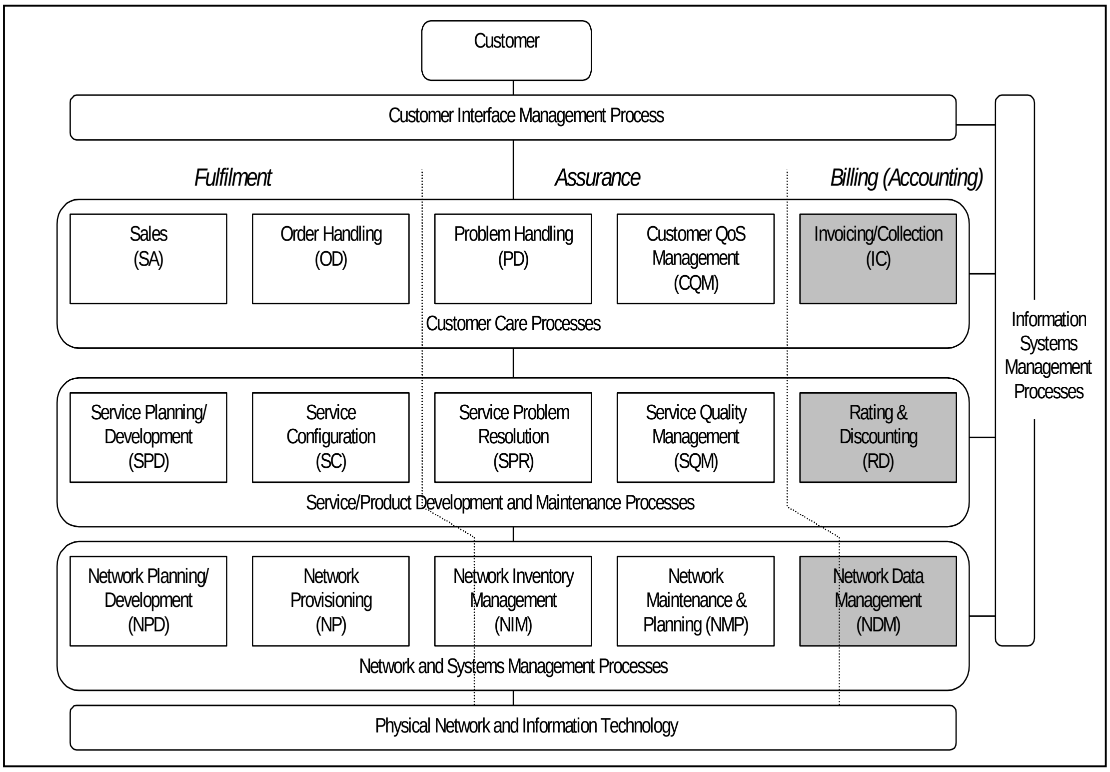


Figure 5.18, The TMF Telecommunications operations map [TMFBoM98]

Vertical process groupings identify three service management areas, Assurance, Fulfilment and Billing. The accounting system implemented an instantiation of the ‘Billing’ business process, which adequately described one important aspect of the system but was renamed ‘Accounting’ for the purposes of the project to take alternative service accounting schemes into consideration. The identified business processes were mapped onto the TINA business roles and used as an analysis basis for demonstration system development (the mapping is shown in Figure 5.19, the TINA business roles are outlined in Figure 4.1). The figure shows the following mappings relevant to accounting [FTD3]:

- invoicing and collection, seen as the responsibility of the Retailer,
- Rating and discounting, seen as the responsibility of the Broker, Retailer and 3rd Party Service Provider,
- Network Data Management, seen as the responsibility of the Connectivity provider and the 3rd Party Service Provider.

As the system was to use service and network level accounting, the process identified as ‘Network Data Management’ in the TMF map was renamed to ‘Usage/Performance Data Collection’ to better signify its role in both the service and network layers.

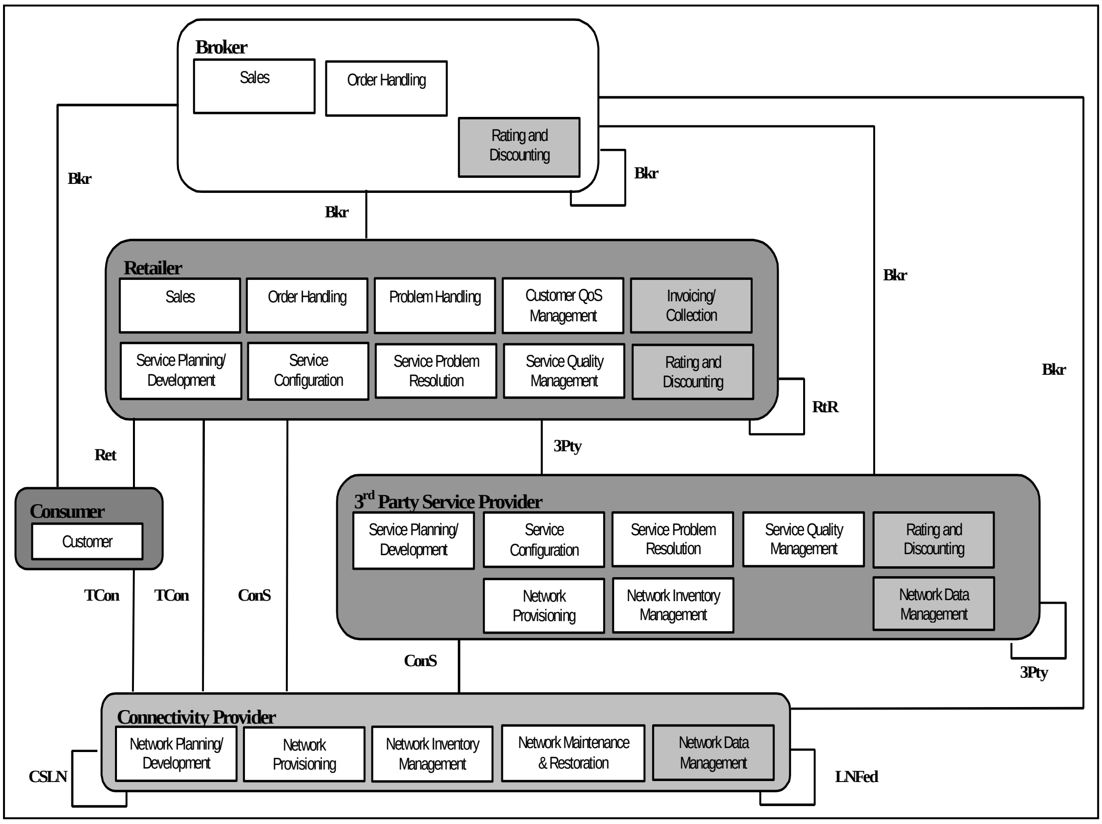


Figure 5.19, Mapping of TMF business processes onto TINA business roles [FTD8]

An overall accounting business model was then elucidated to aid analysis by identifying stakeholders and their roles, this is shown in Figure 5.20.

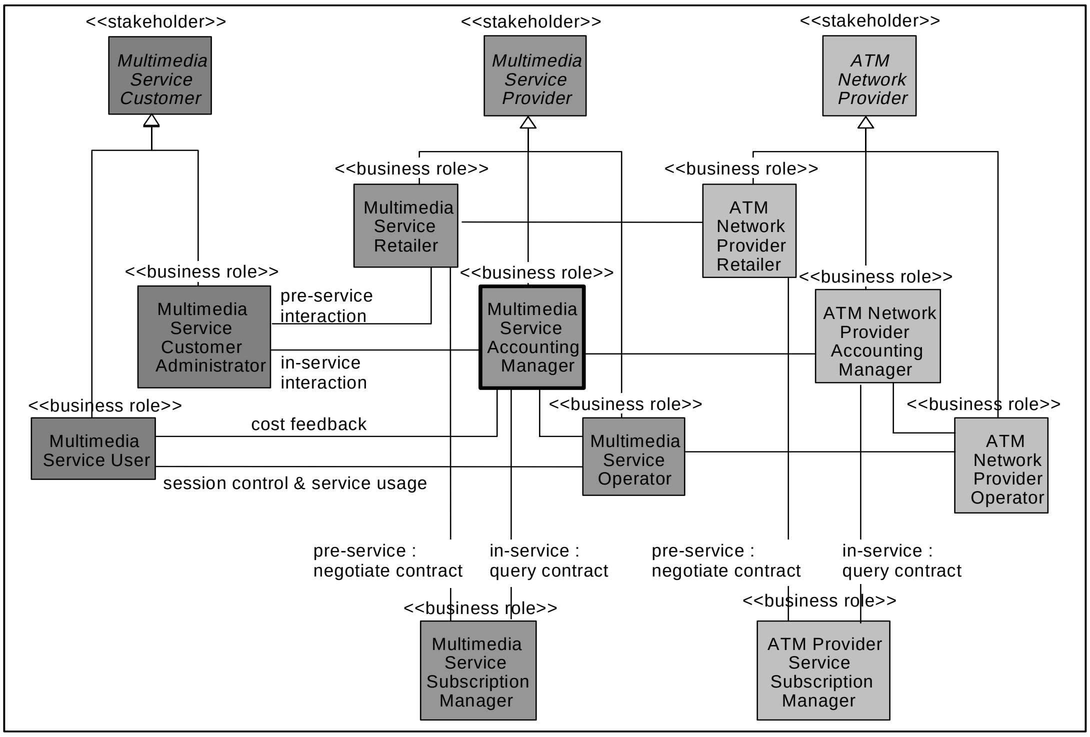


Figure 5.20, The overall accounting system business model

As shown, the three primary stakeholders are, as before, the customer the service provider and the connectivity provider (shown here as an ‘ATM Network Provider’), but the business roles are made more explicit and a split between interactions that occur in the ‘pre-service’ and ‘in-service’ phases are highlighted.

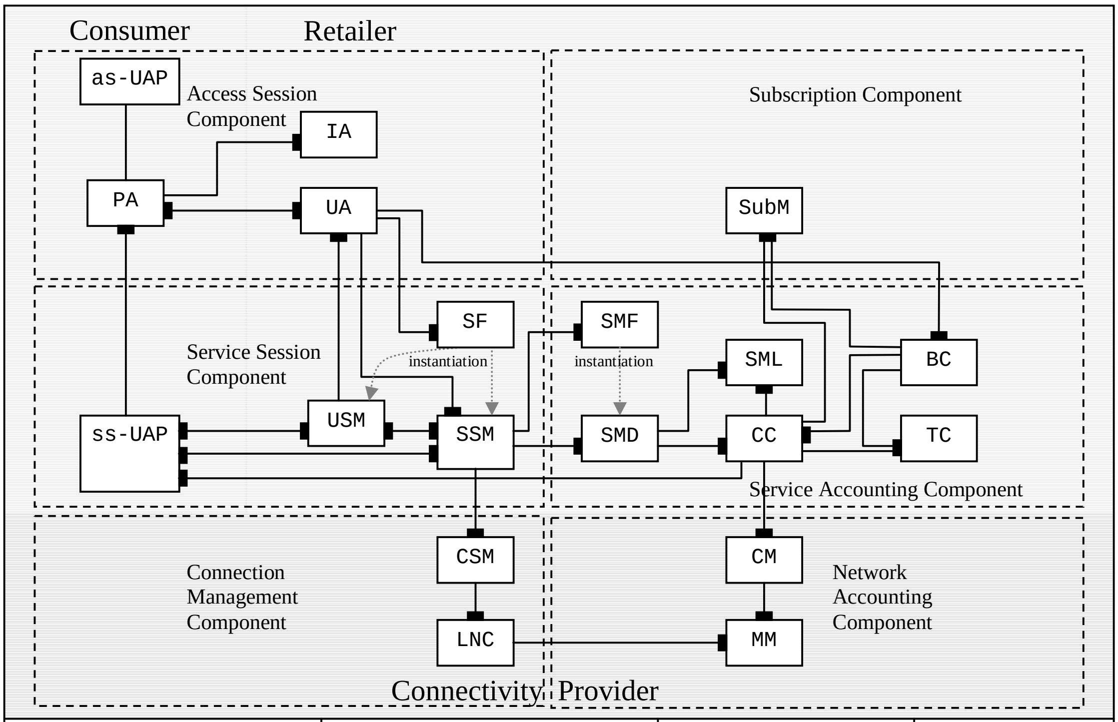
The ‘Accounting Business System’ consisted of components, based on TINA architectural concepts, originally developed in four projects:

- Prospect: service level accounting,
- VITAL: access and service session control (including sample services),
- SUSIE: network level (ATM) accounting, and
- ReTINA: network configuration and connection management.

The specific scenario adopted focused on aspects of the accounting process for the consumer, the connectivity provider, and the invoicing (but not collection) part of the retailer. The following section outlines the implementation of this system.

5.3.2 Project implementation overview

An overview computational model of the system is shown in Figure 5.21.



Abbreviations:		ss-UAP:	service session User Application	LNC:	Layer Network Co-ordinator	SML:	Session Meter Log
		USM:	User (service) Session Manager				
as-UAP:	access session User Application	SSM:	Service session manager	MM:	Metering Manager	CC:	Charge Control
PA:	Provider Agent	SF:	Service Factory	SubM:	Subscription Manager		
UA:	User Agent	CSM:	Communications Session Manager	SMF:	Session Meter Factory	BC:	Bill Control
IA:	Initial Agent	CM:	Charge Manager	SMD:	Session Meter Data	TC:	TariffControl

Figure 5.21, An overall accounting system computational model

In this model there is no separate third party service provider; its functionality is subsumed in the retailer. It is thus the retailer alone that is responsible for the collection of service usage data. The components shown have the following responsibilities:

- The Access Session component was responsible for allowing secure access to subscribed services.
- The Service Session component was responsible for the actual usage of a specific service by a Consumer. In Flowthru it was specialised to cater for two different service paradigms: a multimedia information retrieval service: the ‘Digital Library’, and an interactive multimedia service: the ‘Desktop Video Audio Conference’.
- The Subscription component was responsible for the management of subscription contracts between the Consumer and the Retailer and ensuring access to services is in accordance with these contracts.
- The ATM Accounting component was responsible for collecting network level usage data, generating charges based on this data and forwarding these charges to the Accounting component.
- The Connection Management component was responsible for providing ATM connectivity between end-user equipment and generating network level usage data for the ATM Accounting component.

The ‘Accounting Business Process’ demonstration initially depended on the integration of these components from the several contributing projects. To facilitate this, the objects that were available to other components or that needed to access other component’s objects were highlighted in an overall system component diagram which provided the basis for the diagram above.

The access session components (PA; Provider Agent, UA; User Agent) could establish a secure context for interactions between the Consumer and the Retailer to allow access to subscribed services, where contract details were managed by a subscription component in the Retailer (SA; Subscriber Agent, SubM; Subscription Manager). The service session components (SSUAp; Service Specific User Application, SSM; Service Session Manager) enabled actual service use by a Consumer. These components were specialised to cater for two different service paradigms: an information retrieval service, the ‘Digital Library’, and an interactive service, the ‘Desktop Video Audio Conference’. In Flowthru the SSM was altered to generate the accounting events needed for service level accounting. (The functionality of the TINA Access and Service session management components, along with the Connection Management component were described in detail in section 3.3.3.3.)

The network level (ATM) accounting components (MM; Metering Manager, CM; charge manager) were responsible for collecting network level usage data and generating charges based on this data for the Connectivity Provider, then forwarding these charges to the Retailer accounting component. The Connectivity Provider's configuration and connection management components (CSM; Communication Session Manager, LNC; Layer Network Co-ordinator) were responsible for providing ATM connectivity between end-user equipment and for generating network level usage data for the network level accounting component.

The service accounting component was to be used to account and charge for both service paradigms, combine these with network based charges to produce a bill and enable real time charging with feedback to desktop displays. They are detailed in the following section.

5.3.3 Accounting component implementation

The service accounting component used in Flowthru was based on a similar component developed in Prospect. Although Prospect's architecture was TINA based, it made some simplifying assumptions to avoid over-complicating the prototype software needed for end-user trials. Among these simplifications was the dropping of service independent group session functionality -normally implemented in the SSM- to allow it to be implemented in a service specific way for the services that required it, like the MMC video conferencing service. This was a valid approach as Prospect was mainly concerned with inter-domain service management, not a validation of the TINA architecture, and its IP based services did not need explicit connectivity management that required the SSM to support service session to communication session mappings.

As a consequence of the lack of group session support, the accounting components developed in Prospect were based on single user service sessions; a meter created to record each user's service usage was sufficient to account for a service session. In Flowthru, however, the service components implemented full TINA session functionality which required an alteration of the basis of usage meter creation; each usage meter needed to be associated with a particular service session. The UMData, or Usage Metering Data, computational object within the Prospect accounting component was modified to support the service session as the primary accountable entity for a service, rather than the user, as it had been in Prospect. (A computational object is a unit of distribution in TINA, similar to a CORBA server). To highlight the change in the basis of accounting, the computational object was renamed the Session Metering Data, or SMData.

To facilitate real-time service charge monitoring, the ChargeControl computational object was modified to enable it to notify each user participating in a session of their costs; this also required access to the service accounting events as they were gathered in the SMData. The diagrams that follow show the main service level interactions between the access and service session component computational objects and the service accounting component computational objects used to facilitate service level accounting, charging and billing in Flowthru.

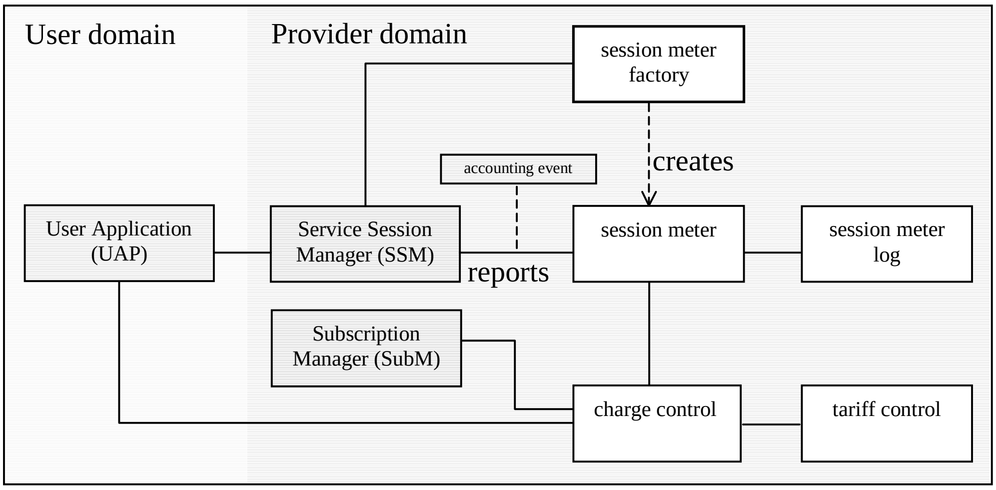


Figure 5.22, 'In Service' accounting component associations

Figure 5.22 shows the interactions between the computational objects primarily involved in service level accounting during actual service use. The shaded computational objects represent computational objects from the service session component; the UAP is the end user's application which allows access to the service functionality implemented in the SSM, the service session manager.

At service session creation, the SSM creates a session meter using the session meter factory in a manner analogous to its own creation by the service factory (SF). As parties join the service session, the session meter is informed and adds parties to the parallel 'accounting session' corresponding to the users they represent. (A user may join a session many times; each user presence is represented by a party and each party can, but need not, play a different role. For example a user could be represented by two parties in a session, one party with a teacher role and another with a student role; each role is charged appropriately.) The user can also register a 'listener' to enable the charge control computational object to feed back charging information to their desktop in real time.

As the user actually uses the service, the SSM generates accounting events that serve to report service usage to the session meter; these events are recorded in the SMData and made available to the charge control. The charge control uses these, along with the user's tariff for that service, stored in the tariff control, as a basis for real time charge calculation enabling subscribed users to 'listen' to their charges. The service can also report overall demand for certain service functionality (be that a physically or information resource based, e.g. how busy a server supporting a service is, or how popular an information resource is) to support demand based pricing if the user has chosen a tariff that enables this.

As the users leave the session, their accounting events are stored, with other session context information, in the session meter log. These persisted events are the basis for 'post-service' accounting component functionality, the associations shown in the figure below.

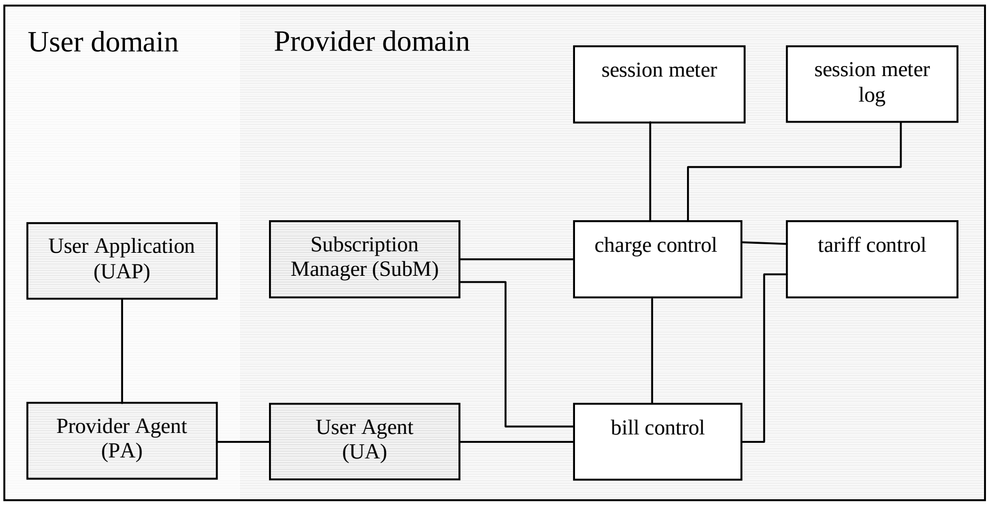


Figure 5.23. 'Post service' accounting component associations

The shaded computational objects represent computational objects from the access session component; the User Agent represents the user in the provider domain, the Provider Agent represents the provider in the user domain and the UAP is the user application, in this case a client capable of requesting a bill and starting service sessions.

When a bill is requested via the User Agent, the bill control computational object requests the charges for that subscriber from the charge control computational object. It then calculates the charges in the same way as it did in the 'In session' scenario presented above, but this time it also bases charges on persisted session information from the session meter log. When the charges are calculated overall subscriber discounts and recurring charges are added by the bill control based on a subscriber specific tariff.

5.4 Summary

This chapter discussed an incremental application of the accounting architecture described in the previous chapter, detailing service accounting component implementation and integration in the context of two projects, Prospect and Flowthru. The implementation work carried out within these two projects resulted in two architectural prototypes, with second prototype, developed as part of this research within Flowthru, benefiting from the lessons learnt in producing the first. The next chapter evaluates the implementation work within Flowthru, highlighting architectural issues brought to light through prototype building and discussing the use and performance of the accounting component.

6 Evaluation and contribution

6.1 Introduction

This chapter evaluates the accounting component developed as part of this research within the Flowthru project, the implementation of which was described in the previous chapter (section 5.3). The evaluation is based on how well this instantiation of the architecture adhered to the architectural principles described in section 4.4, and how well it met the architectural requirements motivating these principles, outlined in section 4.3. (The requirements themselves were derived from an evaluation of the state of the art in telecommunications pricing and software architectures presented in chapters 2 and 3 respectively).

Section 6.2 considers the technology platform used in the implementation, describing the techniques used and how they assisted or frustrated the application of the first principle that the architecture be component centred and DPE based. In so doing, it also considers the appropriateness of a CORBA platform for the implementation of TINA based systems in general, outlining areas of perceived technological deficiency.

Section 6.3 contains the evaluation of the implementation itself, i.e. how it met the other requirements described in section 4.3. These stipulated that an implementation of the architecture should support easily defined and extensible tariffs; service based charging; and feedback mechanisms sustaining adaptive per-user tariffs to enable real time, demand-based pricing. It demonstrates that an implementation which applies the architectural principles outlined in section 4.4 can meet these requirements. Section 6.4 then places the implementation within the context of recent research work. Section 6.5 then presents this research's contribution to the state of the art, before a brief commentary in section 6.6.

6.2 Platform evaluation

The TINA consortium specified components as part of the definition of each sub-architecture, including the service architecture used as a basis for this research. These specifications, when available, were originally given in the consortium's own 'Object Definition Language' (ODL), which was designed to be a superset of the Object Management Group's (OMG) interface definition language (IDL), with additions to deal with multiple interface objects and stream interfaces suitable for "distributed systems in the telecommunications industry" rather than CORBA's "relatively simple architecture and object model" [Kitson95].

The TINA object model allowed objects to support multiple interfaces, and also allowed them to be grouped together to form "building blocks" [Kitson95], component like units of activation and distribution. The presence of these concepts in ODL probably owed their origin to early TINA architectural validation work using an ANSAware platform in the TINA Auxiliary project 'PLATyPus', whose "experience with ANSAware's IDL [provided] valuable insight into how to extend CORBA IDL to satisfy TINA requirements" [Keck95]. These concepts were present in the ANSA architecture, which itself both influenced and drew on RM-ODP concepts and standards [Herbert94].

As CORBA grew in market strength and acceptance, PLATyPus developed an ODL/C++ mapping which extended "the IDL C++ mapping in ways which may readily be implemented using the underlying CORBA platform to support the more stable and well understood aspects of ODL" [Kitson95]. The 'more stable and understood' aspects alluded to TINA operational interfaces, which mapped onto standard IDL interface definitions. Thus PLATyPus, and the TINA validation projects Vital [VD1197] and ALCIN [Hellemans96b] based their architectural implementations on commercial ORB platforms (Orbix and NEO respectively), while streams (i.e. continuous media flows) were dealt with in a project specific manner. (Another TINA validation project, ReTINA, attempted to define a TINA ORB that supported streams and non-implicit binding while endeavouring to "extend the CORBA specifications in an upward compatible way" [Tran96]; components developed within ReTINA were later re-used on a CORBA DPE within the Flowthru project.)

Given that several projects supported by the TINA consortium, some of them actually containing TINA members, chose to use CORBA, and commercial ORB implementations, as a suitable distributed processing environment for the implementation of TINA prototypes, the use of a commercial ORB implementation (Orbix) to provide component inter-operability for this research was not unreasonable. CORBA platforms could easily accommodate the TINA object model, albeit normally at the implementation rather than specification level; for example multiple-interface objects could be supported by multiple inheritance (with C++) or explicit delegation (using 'TIE' objects in Java or C++). Grouping of interfaces was achieved by co-opting the IDL keyword '`module`', `intended` for namespace demarcation, and using it as a means to group operational interface definitions to define components.

While the production of interface specifications guaranteed interoperability, it did not necessarily provide a basis for software re-use. TINA addressed this in the substance of their architectural specifications, dictating that certain components were 'generic', i.e. suitable for use with any service, while service specific aspects could be catered for using inheritance.

This provided for source code level software re-use, but did not allow run-time re-use of the components for multiple services, which is considerably more difficult to achieve, but a more useful form of software re-use. The design of the components for this research considered such run-time re-use for multiple services as one of the requirements to support the rapid development of new services. Such run-time re-use would allow the straightforward inclusion of accounting functionality for newly defined services.

The accounting architecture implementation developed as part of this research within the Flowthru project demonstrated such run-time re-use of all the usage data collection, charging and billing components of the architecture for two services in a public demonstration on a TINA testbed in Alcatel, Antwerp in December 1999. The services, a video conferencing service, the 'Desktop Video Audio Conference' (DVAC), and a digital library service 'DigLib', were implemented in previous projects and needed minor changes to generate the information necessary to enable accounting. The same accounting components were used, without alteration or specialisation, for both services, and also supported the aggregation of network level charges (deemed necessary by the project) in the final bill.

While good specifications facilitate component based re-use in the presence of interoperability guarantees from distributed processing environments like CORBA, they do not enable a market for such components in any realistic sense; there is still a certain amount of information about the components that must be communicated by other means. In TINA the concepts necessary to understand and use the components specified by the architecture were defined in architectural documentation. The 'session' concept defined within the TINA architecture provided the basis for the multi-service run-time re-use of the accounting components created as part of this research (this is described in detail in the next section). Several components developed as part of this research were also used in conjunction with a workflow management system [Wade00], where the process of producing a time delimited bill for services furnished to a subscriber was specified and scheduled using workflow technology. The fact that the components were capable of being used with a separate interaction paradigm demonstrated their flexibility and generality, and also attests to their modularity, in that they fitted easily into an activity centric model. [Fuller00] details the workflow management system and describes its use of the accounting components.

A CORBA component model has been defined to allow easier re-use of CORBA components [RFP960612] in partial recognition of weaknesses in the CORBA architecture with regard to component specification; for example listing the events that a component is capable of receiving and emitting as well as the operations it is capable of carrying out. However, the lack of a robust commercial implementation means that it is being eclipsed by a language specific component model (Enterprise Java Beans, EJB) in the market place.

This research used CORBA services when they were available and their implementations were robust enough to support the reliability requirements of the research projects in which they were used. In particular, the naming service was used to achieve true location transparency. A commercial implementation of the event service became available in the course of this research but proved unreliable, in particular the facility for typed events. As a consequence, where events are specified in the design they were implemented using one-way calls; in reality this does not deviate significantly from a single source/sink event based model, especially when it is borne in mind that the event service specification is itself based on such one-way calls, albeit within a multicast facility. While the inability to have multiple event sinks did restrict the ability to redundantly deploy usage metering facilities, which might restrict any commercial deployment, it was felt that this was an excusable compromise for a prototype system.

Overall, the use of a CORBA platform, together with TINA architectural concepts, can be seen to have provided the capability to define components that were suitable for re-use by many services. Whether the actual components facilitated such re-use is considered in the following section.

6.3 Implementation evaluation

This section evaluates how well the accounting components developed as part of this research adhered to the architectural principles outlined in section 4.4 - these included: the adoption of a session based service architecture, tariffs that were extensible without sacrificing interoperability, a clear separation between a service and the accounting mechanisms for that service, and the use of real time feedback mechanisms to facilitate demand based pricing. It then considers how the observance of each principle helps meet the requirements established in section 4.3 - these included re-usable components, easily defined and extensible tariffs, service based charging, and feedback mechanisms sustaining adaptive per-user tariffs to enable real time, demand-based pricing.

Figure 6.1 shows the main relationships between the architectural principles and the requirements they were intended to support. While the previous section explained how a DPE platform was capable of supporting re-useable components in conjunction with a service architecture, this section also considers how the other relevant principles helped to fulfil this requirement.

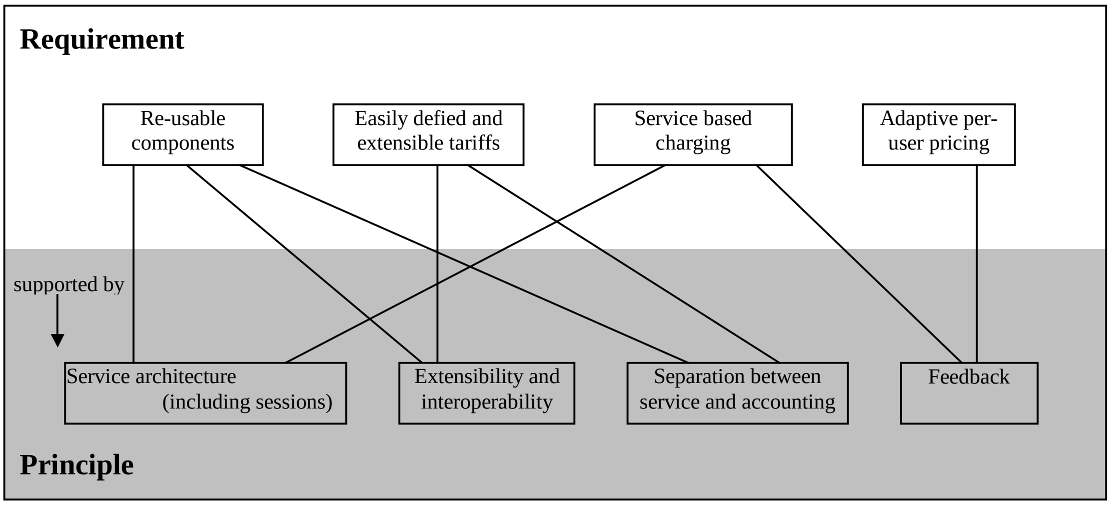


Figure 6.1, A mapping between the architecture's requirements and its principles.

Amongst the most important of the concepts endorsed by the TINA service architecture is that of the 'session' as a context for interactions with a service. As discussed in section 3.3.3.3, several session types were defined because the separation promoted independence between different aspects of service usage. This independence meant that "the modification of a particular session model or related mechanism [did] not impact on models and mechanisms governing other types of session." [Pavon96]. The definition of a separate accounting session would insulate accounting functionality from service specific mechanisms in the similar way. For this reason the accounting architecture adopted the session concept and defined a separate 'accounting session', mirroring the lifecycle of the service session, to provide separate service usage context for accounting purposes. (In this way the principle of separation between service and accounting for that service was also observed; this is discussed further when this principle is considered.)

The model of the session used for accounting purposes is one of the primary differences between the prototypes developed within the Prospect and Flowthru projects. The second architectural implementation supports a generic multi-party, multi-user session, whereas the first only supported single user sessions.

The 'accounting session' is supported by the 'session meter factory' and 'session meter' developed as part of this research within Flowthru, shown in Figure 5.22. These are implemented as separate CORBA objects, but are part of the same CORBA server. A session meter is created for each session. As users are added to the service session, they are also registered in the session meter. This results in the creation of an object to represent that user within the session meter. Under certain circumstances, the same user may join the session more that once, (for example a lecturer making a presentation might like to see how it looks to students). To support this, the session meter also supports multi-party sessions, where a party represents an instance of user participation in the service session. This is modelled by adding another party object to the object representing the user in the session (the first time a user joins a party object is created by default). Thus, in each accounting session the same user may be represented by many parties. Each party can also be assigned a time delimited role which can affect the charging basis. This ability was not used in the Flowthru demonstrations, as this was effected by sequentially creating parties with different initial roles, nevertheless the ability to change role mid-session was thought necessary to enable the capabilities of any TINA service session to be accounted for.

The adoption of the TINA service architecture concepts, including that of the session, helped to fulfil the requirement for service based charging. The service session permitted a resource independent view of service usage which was necessary for service based charging, but not sufficient; the basis for the establishment of the session and the basis for the production of charges must also be taken into account when charging for service usage. For this reason a separate accounting session was defined. This accounting session, while separate from the service session, is nevertheless dependent on it and mirrors its lifecycle, but it includes charging information, including the tariff, which is not an innate part of a service session. The definition of a separate accounting session also facilitates greater software re-use, in that a service no longer needs to build separate accounting functionality into its ‘Service Session Manager’.

The need for extensibility and inter-operability are often conflicting. As a recognition of the need to compromise between these needs in any implementation, they are conflated to form one architectural principle. Extensibility is a broad goal; in section 4.4 it was concerned with the means to account for many services without resorting to service specific mechanisms which would hamper re-use of the components. Where newer versions of a service support more functionality, the tariff and charging mechanism must still support the calculation of charges for older service implementations; the inter-operability between the older service version and the new charging mechanism must be maintained. This is a form of provision for legacy services. Allowing tariffs for newer services to be based on those for older services also facilitates the separation of service logic updates and the updating of its charging basis.

The overall tariff information model is shown in Figure 4.3. The tariff was implemented as outlined; the step between specification and implementation was carried out by an IDL compiler. Extensibility was brought about by basing the tariff on two charge lists, by their nature extensible, which allow calculations to be based on several accounting events and the time interval between any two. This scheme allows tariffs for newer services to be based on those for older services. Inter-operability is ensured by having each service register tariffs with the ‘Tariff Control’. As well as outlining a pricing basis, each tariff specifies the set of possible accounting events the accounting components can expect to receive for a particular service. It is possible to have separate tariffs for each user.

The accounting events in the ‘eventChargeList’ can bring about a once off charge of an amount falling between the maximum and minimum range of charge, where maximum and minimum prices would often be regulated to discourage under- or over-pricing. (The exact means by which the charge can be calculated is discussed when the principle of feedback is discussed.) The same event can also be used as a starting event in a number of concurrent duration based charges with different timing intervals, again with the charge amount falling within a dictated range. This is brought about by making an entry in the ‘durationChargeList’.

A combination of event and duration based pricing can be used to model or approximate the tariff types shown in Figure 2.1. A tariff with a single event based charge is capable of modelling a fixed fee tariff, while a tariff with a single duration based charge is capable of modelling a linear tariff and a tariff with a single event based charge with a duration charge based on the same event is capable of modelling a two-part tariff (and a three part tariff, given a non-unity time interval).

Multi-part block tariffs can be used to approximate smooth tariffs by overlaying several duration based tariffs with different duration intervals, in this case the minimum limit could be negative for some intervals, which would necessitate careful tariff design, lest users be given the ability to make money from using a service! A charging mechanism should not automatically preclude such a possibility though; perhaps some services might want to reward some users that serve to attract others, it might also be the case for early adopters or service testing. Standard non-adaptive peak load pricing can be accommodated by providing a number of tariffs reflecting the average loading periods in a time interval. This would reflect the economic view of, for example, evening and day time telephone usage as that of separate services with common facilities, as only the tariffs would differ.

Overall the implementation ensures tariffs can be extended without affecting inter-operability between different versions of the same service and their common charging basis. The tariffs are defined by specifying the set of ‘accounting events’ that a service can generate with suitable duration and event based price ranges, and more events and duration based charges can be added to extend the tariff without causing problems for previous service versions. The charging model could be criticised for the exclusive use of duration as a charge basis for disparate services, i.e. time is the only independent variable in the charging model. The model might have taken into account other commonly used variables, like bandwidth, when calculating tariffs, but it was thought that this would lead to an unnecessarily network centric view of charging. Versions of a service operating at a different bandwidth can have separately defined tariffs instead. This greatly simplifies the charging model; in effect the accounting components view them as different services. This is consistent with the view that different services cater to different market segments; greater bandwidth is normally synonymous with a higher quality of service in the uncertain world of packet based services, and higher quality services are generally priced appropriately. This view is also consistent with price based rationing schemes for packet networks like 'Paris Metro Pricing' proposed in [Odlyzko98]. The subscription model used in the Flowthru demonstration (shown in the top part Figure 4.3) supported the grouping of tariff identifiers to facilitate such 'Service Level' pricing.

The principle of a separation between a service and the mechanism accounting for that service has been touched upon when discussing the choice of a session basis for charging. This separation, and the existence of a separate 'accounting session', is made possible in the implementation by the definition of separate 'accounting events'. These events, the basis of charging, are published in each service's tariff, and stored for each in the 'Tariff Control' accounting component. During service usage, the accounting events are generated by the TINA 'Service Session Manager' and forwarded to the session meter corresponding to the session. From there they can be persisted in the 'Session Meter Log' during, or at the end of, a service session. The principle of separation between the service and its accounting facilitates software re-use by allowing the same accounting components to be used for many services, the separation also ensures that tariffs can be specified in a service independent manner.

Real-time feedback of pricing information to users is always useful, and is necessary if the price of using a service can change in real-time, as it could under an adaptive pricing regime. Figure 4.2 shows how the accounting components support the feedback of service usage costs to users. This feedback is achieved through a 'chargeListener', a desktop client that renders charging information in an informative manner. If a user chooses to receive pricing information, they register a CORBA object that then becomes the sink for charge information based on that user's usage of the service within the service session. The charge is calculated in real-time by the 'ChargeControl' which then sends the information to the registered 'listener'. It is based on the occurrence of an accounting event interpreted within the context of the accounting session. Per-user pricing was supported by allowing tariffs to be defined for each user of a service, although if necessary the same tariff definition can be shared by many users.

Pricing based on service demand is supported by the service explicitly signalling its load, thereby putting the onus on the service to signal how its usage pattern should affect charging. The figure given is used, with a per-user ‘loadAverageMultiplier’ (present in that user’s tariff; see Figure 4.3), to calculate the actual price for an instance or duration of service usage for a user. The size of the ‘loadAverageMultiplier’ thus signifies the extent to which pricing is affected by the service load; a figure of zero means that pricing is independent of load for that user. The price must fall within the range specified by the ‘chargeRange’ specified as part of that accounting event’s entry in the tariff.

While being adaptive in a limited sense, it could be argued that reducing something as complicated as a switch busy hour to a single figure is somewhat of an oversimplification. Nevertheless such an aggregate figure is possible and could reduce non-price rationing (e.g. delayed, or non-existent dial tone at busy times) by forcing price sensitive users from the system, and, as such, it would be ‘fairer’ in the economic sense. Such adaptive pricing needs the ability to inform the user of the change in the charge basis in real-time; this was possible in the implementation. The adaptivity possible is quite limited and does not facilitate the sort of demand based sensitivity necessary for information based services, which need to be able to price differently at the granularity of a service accounting event, but such monitoring is not impossible within present functionality.

Overall, it can be seen that the architectural principles have been applied in the implementation of the Flowthru accounting components; the TINA session based service architecture was used, with the TINA ‘Service Session Manager’ producing the accounting events necessary to sustain a separate accounting session mirroring the lifecycle of the service session, but independent of it, demonstrating a separation between a service and its charging mechanism. Extensible tariffs are capable of being defined based on those for older services, but preserving inter-operability and price feedback mechanism was implemented to facilitate demand based pricing. The application of these principles was also seen to have supported the requirements that the accounting implementation be DPE based and component centred, and that it support easily defined and extensible tariffs, service based charging, and real time feedback mechanisms to assist the application of adaptive per-user tariffs.

6.4 Context

The accounting architecture proposed was intended to use the facilities provided by an implementation of the TINA service architecture. Thus far this service architecture has achieved little market success, which may appear to diminish the accounting architecture's utility. However, TINA concepts, including the 'session' concept, were shown to be applicable to Internet services in the Prospect project [PD14b98]. Given that an initial implementation of the accounting architecture was based on Internet services, it is not unlikely that accounting architecture could account for them too.

While this research was on-going, there was comparable work which suggested combining service based costs with those of the network, as was actually carried out contemporaneously in the Flowthru project as part of this research [FTD398], [Kausar98]. While Flowthru project considerations necessitated the inclusion of such network based charges, this thesis suggests that they could have been covered by charges at the service layer.

6.5 Contribution

This thesis presented a rigorous examination of the basis of charging for telecommunications service, with the contention that pricing for telecommunications services should be based not on the cost of the underlying communication, but on the value of using the service, where this value can be self-selected, even on an on-going basis. It also showed that present charging mechanisms are still predominantly based on single-service network models that depend on resource level statistics for charging, with misguided, and invariably unsuccessful attempts, to recover similar resource level usage information to provide a charge basis for services on multi –service, packet based, networks.

This thesis proposed, and the implementation demonstrated, that service level, rather than resource level, based accounting is possible. The approach necessitated the selection of the service session as the basic accountable object when dealing with telecommunications services; the service session had already abstracted away the unnecessary details of the underlying technology supporting the service. Accountable events are related based on their presence inside a session, rather than purely historically; there is no need for correlation as the records needed to charge for a specific service usage instance are already related by their origin in a service session. As the notion of the 'service transaction' for which records are correlated is normally defined in a service specific manner, the selection of the session further encourages service independence in the charging mechanism.

To support re-use and independence from the service session, accounting session accounting events are defined separately from service events. This also facilitates significant run time software re-use. The complete description of a service for accounting purposes is contained within its tariff; when that is registered, a session of that service producing the accounting events listed in the tariff is capable of being charged for. No service specific accounting mechanisms are necessary. New service tariffs can be based on successful old tariffs, and services can be modified and tested independently of accounting functionality. The separation between the service and the accounting mechanisms is significant when dealing with new service features, for example it means that current income sources are not jeopardised by testing new services.

This thesis has shown the benefits of service level accounting and has proposed a flexible architecture to facilitate it. This service accounting architecture has been demonstrated in international trials and can be used in conjunction with network level accounting if required [Hellemans99].

6.6 Commentary

The adoption of service level accounting would bring telecommunications accounting away from the resource usage model it has followed blindly since the early days of the telephone's single service network. As multiple new services add new value over and above that provided by the network alone, this value should be capable of being captured and charged for by the network and service providers.

Service level accounting would brings telecommunications accounting to a level above the network technology, a level it should have moved to when the services themselves began to be defined without regard for the underlying network technology. In effect it introduces an accounting abstraction; charging for services at a layer above the usage of the resources that they have been supplied on. This approach inherently allows for greater accounting system component re-use. The service and its accounting mechanism are both built without regard for the underlying network technology. Re-use is further ensured by ensuring a clear separation between the service and its accounting mechanism. With service level accounting it is likely that the processing overhead for bill calculation, traditionally based on correlating umpteen resource usage records corresponding to a service specific 'service transaction', could be considerably reduced.

Service based accounting does not entail a more 'sloppy' approach to accounting; the charges at the service level will need to be carefully calculated to ensure that they can cover any costs they incur, especially those at the network level if there is a separate network provider. Service tariffs would probably be initially framed in conjunction with, or at least with reference to, the network provider who could then take a portion of the charge. Perhaps perversely, it will probably be the network provider who will be in the best position to provide the software infrastructure necessary to account at the service level. For a price. Service providers could introduce any number of new services which, with a flexible accounting mechanism furnished by the network provider, they could then charge for. The problem of how to exclude non-paying services would then have to be addressed: there could be technical solutions which build on Cox's 'meterware' ideas [Cox96], or solutions based on protected service specific features, but blunter strategies seem to underlie the recent convergence of content service and network providers. The danger with such convergence is that it may create closed networks which limit new service innovation, but whatever the services available on these networks, they could still use service level accounting schemes.

The introduction of service based charging would involve some political will too; the introduction of any new accounting scheme leaves potential for abuse. It could be all too easy to set prices at far above cost in a service level accounting scenario, for this reason there would be a need for continued regulatory vigilance in the absence of true competition.

7 Conclusions

7.1 Summary of research

This research examined the price basis and mechanisms for disparate telecommunications services, outlining requirements that any system accounting for these services should meet. It examined several telecommunications software architectures in historical progression appraising their abilities to support a generic re-usable accounting mechanism. It also examined the actual provision made for accounting in these architectures.

It found that two of the architectures considered, those of the ‘Intelligent Network’ and the ‘Telecommunications Management Network’ were incapable of supporting such an accounting architecture as, overall, the first failed to sufficiently consider operations support for new services, while the latter failed to support service creation. The specific provisions for accounting in these architectures was also found to be lacking. The third software architecture, that of the ‘Telecommunications Information Networking Architecture Consortium’, was considered sufficient to support a generic re-usable accounting mechanism, but the specific provisions made for accounting in this architecture had several shortcomings including service dependent, resource based, charging mechanisms.

An architecture to support the requirements for re-usable components, easily defined and extensible tariffs, service based charging, and adaptive, per-user pricing, was then suggested. This was based on the principles of service session based accounting, extensibility and inter-operability, the separation of service and accounting and price feedback.

The architecture was then applied. Its incremental application for four separate services, within the context of two European research projects was documented. The implementation work was assessed with regard to its adherence to the architectural principles, and then with regard to how the application of the principles helped it to meet the original requirements. The assessment was also divided into an assessment of the platform support necessary for the architecture and the components of the architecture itself.

7.2 Conclusions from research work

This research suggests that an accounting mechanism operating at the service layer can recognise and charge for the added value that new services offer over and above basic connectivity.

It suggests that the traditional fixation on producing, aggregating and interpreting network level measures in an attempt to account for service usage fails to address the perceived value in using a service, and overlooks the fact that a software defined service has the ability to produce its own usage information that can be used as a basis for charging. The capability to charge at the service level may become more important with the growth of multi-service networks because network level accounting schemes for these networks fail to address several basic value attribution problems, for example whether the receiver or sender of information should pay for a particular service transaction. It also suggests that a failure to abstract from the underlying network when devising a charging scheme has repercussions on the ability to provide a service independent accounting mechanism.

The research showed that a service-level accounting architecture was realisable. It also showed that an implementation was capable of meeting the architectural requirements for re-usable components, which was demonstrated by run-time accounting component re-use for two separate services; for easily defined and extensible tariffs, which was addressed by the provision of a generic basis for tariff specification which included extensibility; for service based charging, which was addressed by extending the TINA session concept to define an ‘accounting session’ with separately defined ‘accounting events’ which also facilitated service independence in the charging mechanism; and for adaptive, per-user pricing, which was facilitated by the real-time feedback of pricing information with explicit service load indications.

7.3 Limitations and suggestions for future work

The architecture as suggested suffers from a number of limitations, but the most immediately apparent one is the lack of appreciable market success for the underlying TINA service architecture which this research is based on, and for whose services it would hope to account. However, TINA concepts, including the ‘session’ concept, were shown to be applicable to Internet services in the Prospect project [PD14b98] which included an initial implementation of the accounting architecture as part of this research.

Another problem might be resistance to service based charging if it was mooted in the real world, but with enough competition and good marketing, and especially, with good services, it might be acceptable.

The research's main deficiency lies in its failure to address inter-domain accounting issues while dealing with service value issues. This research focused on the fundamental basis of charges, not their interchange: the means to negotiate the split between service providers, and its basis was not considered. The inter-communication of services whose users originate on separately provided networks was not addressed, nor was the basis for a likely subdivision of service based charges between the network providers. In such a scenario, brokerage services, which can mediate between service providers, offer a direction for future research that might address this shortcoming.

Another deficiency lies in the fact that, while adaptive pricing was possible it could only be provided at a very coarse level of granularity; it would be better if individual service accounting events were capable of price changes to reflect generally perceived trends of service usage across the service sessions of many users. This would also enable information which is sensitive to time and market size to change its price with respect to its age and audience, as well as general service load, in real-time. Future research might address this shortcoming; it may be necessary to specify a separate policy to direct the interpretation of service accounting events in this case, as they be used to alter prices in more ways than was originally intended. The basis for session meter collaboration would also need to be well specified. This policy description would need to be part of the tariff to ensure it remains the sole description of a service for accounting purposes.

The implementation does not record the real-time charges generated and fed back to service users in real-time, nor does it provide a facility to record them. Session state is preserved and re-used to calculate periodic bills; while this is currently acceptable, it does lead to a needless duplication of effort in bill calculation, and if a finer grained adaptive pricing mechanism was adopted it would probably be more sensible to record generated charges than their basis, although it is likely that some recorded session and service state information will be necessary for auditing purposes.

Bibliography

[90/387/EEC] Council of the European Communities. Council Directive of 28 June 1990 on the establishment of the internal market for telecommunications services through the implementation of open network provision. Official Journal of the European Communities No. L 192, 24/7/1990

[90/388/EC] Commission of the European Communities. Commission Directive of 28 June 1990 on competition in the markets for telecommunications services. Official Journal of the European Communities No. L 192, 24/7/1990

[97/114/EC] Commission of the European Communities. Commission Decision of 27 November 1996 concerning the additional implementation periods requested by Ireland for the implementation of Commission Directives 90/388/EEC and 96/2/EC as regards full competition in the telecommunications markets. Official Journal of the European Communities No. L 41, 12/02/1997

[97/51/EC] Council and Parliament of the European Communities. Directive 97/51/Ec Of The European Parliament and of The Council of 6 October 1997 amending Council Directives 90/387/EEC and 92/44/EEC for the purpose of adaptation to a competitive environment in telecommunications. Official Journal of the European Communities No. L 295, 29/10/1997

[Adler95] Richard M. Adler. Distributed Coordination Models for Client/Server Computing. *IEEE Computer*, 28(4):14-22, April 1995.

[Anania97] Loretta Anania and Richard Jay Solomon. Flat – The Minimalist Price. In Lee W. McKnight and Joseph P. Bailey editors, *Internet Economics*, pages 91-118. The MIT Press, 1997.

[Appledorn93] Menso Appledorn, Roberto Kung, Roberto Saracco. TMN + IN = TINA. *IEEE Communications Magazine*, 31(3): 78-85, March 1993.

[Arango93] M. Arango *et al.*. The Touring Machine System. *Communications of the ACM*, 36(1), 68-77, January 1993.

[Barlow94] John Perry Barlow. The Economy of Ideas. *Wired*, 2(3), March 1994.

[BThistory] The Historical Development of BT. Available at http://www.bt.com/archives/history.htm

[BandX97] http://www.band-x.com/about.cfm

[Bangemann97] Martin Bangemann. A New World Order for Global Communications. Speech to International Telecommunications Union, Geneva, Switzerland, September 1997. Available at: http://www.ipso.cec.be/infosoc/promo/speech/geneva.html (1999).

[Barr93] William J. Barr, Trevor Boyd, Yuji Inoue. The TINA Initiative. *IEEE Communications Magazine*, 31(3): 70-76, March 1993.

[Bauer88] Helen A. Bauer, John J. Kultzer, Edward G. Sable. Designing Service-Independent Capabilities for Intelligent Networks. *IEEE Communications Magazine*, 26(12): 31-41, December 1988.

[Berman92] Roger K. Berman and John H. Brewster. Perspectives on the AIN Architecture. *IEEE Communications Magazine*, 30(2): 27-33, February 1992.

[Berndt94] Berndt Hendrik, Peter Graubmann, Masaki Wakano. Service Concepts in TINA-C. In *Proceedings of the International conference on Intelligence in Services and Networks* (IS&N 94), pages 355-366, 1994.

[Black95] Uyless Black. The X series recommendations: standards for data communications. McGraw-Hill, 1995.

[Bleakley97] Bleakley, C., Donnelly, W., Lindgen, A., Vuorela, H. TMN Specifications to Support Inter-Domain Exchange of Accounting, Billing and Charging Information, in *Proceedings of the 5th International Conference on Intelligence in Services and Networks*, Cernobbio, Italy, pages 275-282, Springer-Verlag, May 1997.

[Bodamer98] Stefan Bodamer. Charging in Multi-Service Networks, 1998. Available at: http://www.ind.uni-stuttgart.de/Content/Staff/50_WiMa / bodamer/User/IB-Charging98.pdf

[Brooks75] Frederick P. Brooks. The mythical man-month : essays on software engineering. Addison-Wesley, 1995, previous edition 1975.

[Brownlee97] Nevil Brownlee. Internet Pricing in Practice. In Lee W. McKnight and Joseph P. Bailey editors, *Internet Economics*, pages 77-90. The MIT Press, 1997.

[Bruno99] Lee Bruno, ‘Broadband: to infinity and beyond’, Red Herring Magazine
(ISSN 1080-076X), February 2000, pages 170-171

[Cairncross95] Frances Cairncross. The death of distance. *Economist Newspaper*, London, September 30, 1995.

[Cameron93a] E. Jane Cameron, Nancy Griffeth, Yow-Jian Lin, Margaret E. Nilson, William K. Schnure and Hugo Velthuijsen. A Feature-Interaction Benchmark for IN and Beyond.

[Cameron93b] E. Jane Cameron and Hugo Velthuijsen. Feature Interactions in Telecommunications Systems. *IEEE Communications Magazine*, 31(8):18-23, August 1993.

[Carlton75] Lynne Carlton. Sensitising Telephone rates. *Telephony*, 15(9):66-70,138, September 1975.

[Clark96] David D. Clark. Adding service discrimination to the Internet. *Telecommunications Policy*, (Special issue: Lessons from the Internet), 20(3):169-181, April 1996.

[Clark97] David D. Clark. Internet cost allocation and pricing. In Lee W. McKnight and Joseph P. Bailey editors, *Internet Economics*, pages 215-252. The MIT Press, 1997.

[CNET98] Jim Hu. AT&T Limits Flat-rate Net Hours. Available at: http://www.canada.cnet.com/news/0-1005-200-327944.html, March 1998.

[Collet92] Patrice Collet, Roberto Kung. The Intelligent Network in France. *IEEE Communications Magazine*, 30(2):82-89, February 1992.

[Comptel99] Billing Mediation Solutions. Available at http://www.comptel.com/solutions/billing.asp

[Cox96] Brad Cox. Superdistribution: objects as property on the electronic frontier. Addison-Wesley 1996.

[Cuiker99] Kenneth Neil Cuiker, ‘Pulling in the Net’, Red Herring Magazine (ISSN 1080-076X), February 2000, pages 208-210.

[Davison99] R. Davison and T. Turner. A Practical Perspective on TMN Evolution. In Proceedings of the 6th International Conference on Intelligence in Services and Networks, Barcelona, Spain, pages 3-12, Springer-Verlag, April 1999.

[Dupuy95] Fabrice Dupuy, Gunnar Nilsson, Yuji Inoue. The Tina Consortium: Toward Networking Telecommunications Information Services. *IEEE Communications Magazine*, 33(11):78-83, November 1995.

[Duran92] José M. Duran, John Visser. International Standards for Intelligent Networks . *IEEE Communications Magazine*, 30(2): 34-42, February 1992.

[Dyson95] Esther Dyson. Intellectual Value. *Wired*, 3(7), July 1995.

[Edell95] Richard J. Edell, Nick McKeown and Pravin P. Varaiya. Billing users and Pricing for TCP. IEEE Journal on Selected Areas in Communications, 13(7):1162-1175, September 1995.

[ETSITR101] European Technical Standards Institute. Network Aspects: Considerations on Network Mechanisms for Charging and Revenue Accounting (ISBN 2-7437-2694-6), TR101 619 V1.1.1 , November 1998.

[EUTP99] Status Report on European Union Telecommunications Policy. European Union, Directorate-General XIII. March 1999.

[FTD398] Costas Stathopoulos (ed.). ACTS Project AC335: Flowthru Deliverable 3: System Specifications and Models. November 1998.

[FTD800] William Donnelly (ed.) ACTS Project AC335: Flowthru Deliverable 8: Final Report. January 2000.

[Farley00] Tom Farley. Private Line's Telephone History. Available at http://www.privateline.com/TelephoneHistory/History1.htm. June 2000.

[Ferguson92] Eugene S. Ferguson. *Engineering and the Mind's Eye*. MIT Press, 1992.

[Fishburn98] Peter C. Fishburn and Andrew M. Odlyzko. Dynamic Behaviour of Differential Pricing and Quality of Service Options for The Internet. *In Proceedings ICE'98* Charleston SC, USA. Pages 128-139. ACM Press, 1998.

[Fowler86] Mark S. Fowler, Albert Halprin, James D. Schlichting. Back to the Future: A Model for Telecommunications. *Federal Communications Law Journal*, 38(2):145-200, 1986.

[Fuente94] Luis A. de la Fuente, Juan Pavon, Nikolaus Singer. Application of TINA-C Architecture to Management Services. In *Proceedings of Intelligence in Services and Networks* (IS&N 94), pages 273-284, 1994.

[Fuller00] John Fuller. A Framework for CORBA Component Integration and Invocation in a Workflow Environment. MSc. Thesis, University of Dublin, October 2000.

[Gansert88] Joseph A. Gansert. Intelligent Network Evolution Issues. *IEEE Communications Magazine*, 26(12): 59-63, December 1988.

[Garfinkel75] Larry Garfinkel, Usage Sensitive Pricing: Studies of a New Trend, *Telephony*, 10(2):24-29, 1975.

[Garrahan93] J.J. Garrahan, P. A. Russo, K.Kitami and R. Kung. Intelligent Network Overview. IEEE Communications Magazine, 31(3):30- 38, March 1993.

[Gilmour88] Julia Gilmour and Richard D. Gove. Intelligent Network/2: The Architecture – The Technical Challenges – The Opportunities. *IEEE Communications Magazine*, 26(12): 8-11, December 1988.

[Glitho95] Roch H. Glitho, Stephen Hayes. Telecommunications Management Network: Vision vs. Reality. *IEEE Communications Magazine*, 33(3): 47-52, March 1995.

[Goldhaber96] Michael H. Goldhaber. Attention shoppers! *Wired*, pages 182-189, December 1997.

[Gopal92] Gita Gopal, Gary Herman, Mario P. Vecchi. The Touring Machine Project: Toward a Public Network Platform for Multimedia Applications. In *Proceedings of the 8th International Conference on Software Engineering for Telecoms Systems and Services*, Florence, March 1992.

[Griffeth93] Griffeth, Extending telecommunications systems: The feature-interaction problem, *IEEE Computer* August 1993

[Gupta97] Alok Gupta, Dale O. Stahl and Andrew B Whinston. Priority Pricing of Integrated Service Networks. In Lee W. McKnight and Joseph P. Bailey editors, *Internet Economics*, pages 323-352. The MIT Press, 1997.

[Hall94] Jane Hall, Thomas Magedanz. Uniform Modelling of Management and Telecommunication Services in Future Telecommunication Environments based on the ROSA Approach. In H.G. Hegering, Y. Yemini, editors, *Integrated Network Management III*. Elsevier Science Publishers B.V., 1993.

[Hamada96] Takeo Hamada, TINA Accounting Management Architecture, In *Proceedings TINA'96 Conference* (ISBN 3-8007-2201-1), VDE-VERLAG, Berlin & Offenbach, 1996.

[Hass88] Ronald J. Hass, Stewart D. Personick. Working with the Industry to Shape Intelligent Network Requirements. *IEEE Communications Magazine*, 26(12): 12-15, December 1988.

[Head88] Charles S. Head. Intelligent Network: A Distributed System. *IEEE Communications Magazine*, 26(12): 16-20, December 1988.

[Hellemans96] Patrick Hellemans, A Comparison Between TINA and Legacy Signalling, In *Proceedings TINA'96 Conference* (ISBN 3-8007-2201-1), VDE-VERLAG, Berlin & Offenbach, 1996.

[Hellemans96b] Patrick Hellemans, Piet De Cueleners, Guy Reyniers, Frank Steegmans, Hans Vanderstraeten, Hans Zandbelt. The ALCIN Project: TINA Validation Through Prototyping. In *Proceedings TINA'96 Conference* (ISBN 3-8007-2201-1), pages 125-134, VDE-VERLAG, Berlin & Offenbach, 1996.

[Hellemans99]	Patrick Hellemans, Cliff Redmond, Koen Daenen and David Lewis. Accounting Management in a TINA-Based service and Network Environment. In *Proceedings of Intelligence in Services and Networks* (IS&N'99).
[Herbert94]	Andrew Herbert. An ANSA Overview. *IEEE Network*, pages18-23, January/February 1994.
[Homayoon88]	Seth Homayoon, Harmi Singh. Methods of Addressing the Interactions of Intelligent Network Services with Embedded Switch Services. *IEEE Communications Magazine*, 26(12):42-46, December 1988.
[Huber93]	Peter Huber. Two Cheers for Price Discrimination. *Forbes Magazine*, September 27, 1993.
[Huber97]	Peter Huber. Law and Disorder in Cyberspace: abolish the FCC and Let Common Law Rule the Telecosm. Oxford University Press, 1997.
[Humes88]	Robert W. Humes. Editorial: Special Edition on Intelligent Networks. *IEEE Communications Magazine*, 26(12):7, December 1988.
[Hssworld99]	Hughes Software Systems, http://www.hssworld.com/
[Hwang96]	Woosun Hwang, Eunchul Kim, Heekyung Jeong. Realisation of TINA Service Architecture on the Internet. In *Proceedings TINA'96 Conference* (ISBN 3-8007-2201-1), pages 235-244. VDE-VERLAG, Berlin & Offenbach, 1996.
[Isenberg98]	David S. Isenberg. The Dawn of the Stupid Network. Mixed Media. ACM Networker. February/March 1998.
[Ithell89]	Alan H. Ithell. Service Integration on Digital Networks. Cimtech (Hatfield Polytechnic), 1989.

[Jacobson92]	Ivar Jacobson. Object-oriented software engineering : a use case driven approach. ACM Press, Addison-Wesley, 1992.
[Jacobson99]	Ivar Jacobson, Grad Booch and James Rumbaugh. The Unified Software development process. Addison-Wesley, 1999.
[Kausar98]	Nadia Kausar, Bob Briscoe and Jon Crowcroft. A charging model for sessions on the Internet, 1998.
[Kirzner81]	Israel M. Kirzner. The 'Austrian' Perspective on the Crisis. In, Daniel Bell and Irving Kristol, editiors, *The Crisis In Economic Theory*. Basic Books, New York 1981.
[Kitson95]	Barry Kitson. CORBA and TINA: The Architectural Relationships. *In Proceedings of TINA 95*, Melbourne, Australia. Volume 2; pages 371-386. TINA Consortium/ IEEE Press, February 1995.
[Keck95]	Brian Keck, Barry Kitson, Steve Leask, Ian Lewis, Ajeet Parhar, Peter Richardson and Geoff Wheeler. PLATyPus – Roadtesting and Tuning TINA. In *Proceedings of TINA 95*, Melbourne, Australia. Volume 2; pages 719-733. TINA Consortium/ IEEE Press, February 1995.
[Kockelmans95]	Maarten Kockelmans, Eric de Jong. Overview of IN and TMN Harmonization. *IEEE Communications Magazine*, 33(3):62-66, March 1995.
[LaPorta97]	Thomas F. La Porta and Kuo-Wei Herman Chen. A Direct Signalling System for Flexible Access and Deployment of Telecommunications Services. *IEEE/ACM Transactions on Networking*, 5(4): 489-501, August 1997.
[Laffont00]	Jean-Jacques Laffont and Jean Tirole. Competition in Telecommunications. The MIT Press, 2000.
[Lappin96]	Adrian Lappin. Billing Systems and IN. *Intelligent Networks Conference*, Claridges Hotel, London, May 1996.

[Lengdell96] Magnus Lengdell, Juan Pavon, Masaki Wakano, Martin Chapman, Motoharu Kawanishi. The TINA Network Resource Model. *IEEE Communications Magazine,* 35(3): 74-79, March 1996.

[Lewis97] David Lewis, Thanassis Tiropanis, Alistair McEwan, Cliff Redmond, Vincent Wade, Ralf Bracht. Inter-Domain Integration of Services and Service Management. In *Proceedings IS&N'97,* Como, Italy, May 1997.

[Lewis97b] D. Lewis, T. Tiropanis, A. McEwan, C. Redmond, V. Wade, R. Bracht. Experiences in Integrated Multi-Domain Management. In Proceedings *IFIP/IEEE International Conference on Management of Multimedia Networks and Services*, Motreal, Canada, 1997.

[Lewis98] David Lewis and Thanassis Tiropanis. Integrating TINA into an Internet-based Services Market. In *Proceedings of the International Conference on Intelligence in Services and Networks (IS&N'98).*

[Lewis99] David Lewis, 'A Development Framework for Open Service Management Systems'. In *Interoperable Communications Networks*, 2(1):11-30. Baltzer Science Publications, 1999.

[Li94] Haijiang Li. A component theory for telecommunications Architectures.In *Proceedings of the International Conference on Intelligence in Services and Networks (IS&N'94).*

[Littlechild79] Stephen C. Littlechild. Elements of telecommunications economics. Institution of Electrical Engineers, London., 1979.

[Mackie-Mason95a] Jeffrey K. MacKie-Mason and Hal R. Varian. Pricing congestible Network Resources. IEEE Journal on Selected Areas in Communications, 13(7):1141-1149, September 1995.

[Mackie-Mason97a] Jeffrey K. MacKie-Mason and Hal R. Varian. Economic FAQs About the Internet. In Lee W. McKnight and Joseph P. Bailey editors, *Internet Economics*, pages 27-62. The MIT Press, 1997.

[Mackie-Mason97b] Jeffrey K. MacKie-Mason, Liam Murphy and John Murphy. Responsive Pricing in the Internet. In Lee W. McKnight and Joseph P. Bailey editors, *Internet Economics*, pages 280-303. The MIT Press, 1997.

[M.3010] M.3010. Principles for a Telecommunications Management Network. International Telecommunications Union, 1992.

[M.3200] M. 3200. TMN Management Services and telecommunication managed areas: overview. International Telecommunications Union, 1992.

[M.340092] M.3400. TMN Management Functions. International Telecommunications Union, 1992.

[MaA94a] Mícheál Mac an Airchinnigh and Hans-Jürgen Kugler. Service Engineering Versus Software Engineering – A Foundational Study. In *Proceedings of the International Conference on Intelligence in Services and Networks (IS&N'94),* Springer-Verlag 1994.

[Marshall95] I.W. Marshall, M. Bagley. The Information Services Supermarket – an Information Network Prototype. *BT Technology Journal*, 13(2), April 1995.

[McKinney98] R.D. McKinney, Warren A. Montgomery, Hamza Ouibrahim, Paul Sijbem and John J. Stanaway Jr. Service-Centric Networks. Bell Labs Technical Journal, July-September 1998, pages 98-115.

[Mercury93] Mercury Communications. The Magic of IN. Mercury Communications Ltd. Available at http://www.gare.co.uk/technology_watch/in.htm. February 1993.

[Mierop93] John Mierop, Stefan Tax, Ronald Janmaat. Service Interaction in an Object-Oriented Environment. *IEEE Communications Magazine*, 32(8): 46-51, August 1993.

[Mitchell91] Bridger M. Mitchell and Ingo Vogelsang. Telecommunications Pricing: Theory and Practice. Cambridge University Press, 1991.

[MitchellR97] Russ Mitchell. Eli Noam Wants to Disturb You. *Wired Magazine*, 5(10), October 1997. Available at http://www.wirednews.com/wired/archive/5.10/cyberspace_pr.html

[Mowbray95] The Essential CORBA, Thomas J. Mowbray & Ron Zahavi

[Mueller97] Milton Mueller. Myth Made Law: Universal Service and the Telecommunications Act. Communications of the ACM, 40(3):39-47, March 1997.

[Mäkelä99] Teemu Mäkelä. Accounting in the PSTN. Available at http://www.tml.hut.fi/Opinnot/Tik-110.551/1999/papers/09AccountingInPSTN/accounting.html, April 1999.

[Nat92] N. Natarajan, G.M. Slawsky. A Framework Architecture for Multimedia Information Networks. *IEEE Communications Magazine*, 31(2): 97-104, February 1992.

[Netman94] RACE H408. Accounting Management Services in TMN. Available at ftp://ftp.uni-stuttgart.de/pub/org/CFS/race-cfs/rh408.doc.gz

[OMG99-02-05] Object Management Group. RFP: 99-02-05. CORBA Component Model.

[Odlyzko97] Andrew Odlyzko. A Modest Proposal for Preventing Internet Congestion. DIMACS Technical Report 97-69. October 1997.

[Odlyzko98a] Andrew Odlyzko. Paris Metro Pricing for the Internet. Available at http://www.research.att.com/~amo/doc/networks.html

[O'RR98] Maureen O'Reilly Roche. Call Party Handling Using the Connection View State Approach: A Foundation for Intelligent Control of Multiparty Calls. IEEE Communications Magazine, 36(6):60-66, June 1998.

[P226D495] Eurescom Project P226. TMN Management of IN-based Services and Mobility of pan-European IN-based Services. Details available at http://www.eurescom.de/public/projects/p200-series/p226.htm

[P407D197] Eurescom Project P407. Specification of TMN X-interface for Charging, Billing and Accounting. Details available at http://www.eurescom.de/Public/Projects/p400-series/p407.htm

[PD14b98] Ralf Bracht (ed.). Prospect deliverable D14b: The Final System Model. August 1998.

[PD22B98] Vincent Wade (ed.). Prospect deliverable D22b: Prospect Methodology for the design of Multi Domain Management Services in an Open Services Environment (Trial 2). February 1998.

[PD4B98] Siamak Peykarimah (ed.). Prospect deliverable D4: Specification of Inter-Domain Management Services for Second Trial. February 1998.

[Pavon96] Juan Pavon. Towards Integration of Service and Network Management in TINA. *Journal of Network and Systems Management*, 4(3): 299-318, 1996.

[Perez94] Franklin Perez. The Case for a Deregulated Free Market Telecommunications Industry. *IEEE Communications Magazine*, 34(12):63-70, December 1994.

[Pezzutti92] David A. Pezzutti. Operations Issues for Advanced Intelligent Networks. *IEEE Communications Magazine*, 31(2): 58-63, February 1992.

[Pierce88] Michael Pierce, Fred Fromm, Fred Fink. Impact of the Intelligent Network on the Capacity of Network Elements. *IEEE Communications Magazine*, 26(12): 25-30, December 1988.

[Pinkham88] Gary Pinkham, Intelligent Network Feature Transaction Interface. *IEEE Communications Magazine*, 26(12): 47-52, December 1988.

[Pioneer97] James Romenesko. Will the flat rate fly? Available at http://www.pioneerplanet.com/archive/netsurvey/dox/survey2.htm, March 1997.

[Pras99] Aiko Pras, Bert-Jan van Beijnum, Ron Sprenkels. Introduction to TMN. CTIT Technical Report 99-09, University of Twente, April 1999.

[Prepare96] 'Management of Telecommunications Systems and Services: Modelling and Implementing TMN-Based Multi Domain Management'. Jane Hall (ed.). Springer Verlag Berlin/Heidelberg, 1996. (Lecture notes in computer science 1116.)

[Q.82598] Q.825. Specification of TMN Applications at the Q3 Interface: Call Detail Recording. ITU-T 1998.

[RFC1462] E. Krol and E. Hoffman. RFC 1462: FYI on "What is the Internet?" (available at http://info.internet.isi.edu/in-notes/rfc/files/rfc1462.txt). May 1993.

[RFC1958] B. Carpenter (ed.). RFC 1958: Architectural Principles of the Internet (available at http://info.internet.isi.edu/in-notes/rfc/files/rfc1958.txt). June 1996.

[RFC2109] D. Kristol and L. Montulli. RFC 2109: HTTP State Management Mechanism (available at http://sunsite.dk/RFC/rfc/rfc2109.html). February 1997.

[RFP960612] 'CORBA Component Model Request for Proposal', OMG Document orbos/96-06-12, Object Management Group,1996. Available at http://cgi.omg.org/cgi-bin/doc?orbos/97-06-12.pdf

[Ranasinghe94] Donald Ranasinghe, Jørgen Nørgaard. Charging generalised: a generic role model for charging and billing services in telecommunications. In Jarmo Harju, Tapani Karttunen and Olli Martikainen, editors, *Intelligent Networks - Proceedings of the IFIP workshop on Intelligent Networks*, pages 159-172. Chapman and Hall, 1994.

[RateXchange98] http://www.ratexchange.com/

[Redmond97c] Cliff Redmond and Vincent Wade. Dynamic charging for information services.In *Proceedings IEE Colloquium on Charging for ATM*, London, November 1997.

[Redmond98a] Cliff Redmond and Vincent Wade. Towards Flexible Metering & Charging For Information Services. *In Proceedings ICCC/IFIP Conference on Electronic Publishing*, Budapest, Hungary. April 1998.

[Robinson62] Joan Robinson. Economic Philosophy. Penguin books, 1981.

[Ridley97] Matt Ridley. The Origins of Virtue. Penguin books, 1997.

[Rubin94] Harvey Rubin, N. Natarajan. A Distributed Software Architecture for Telecommunications Networks. *IEEE Network*, 8-17, January/February 1994.

[Schenker96] Scott Shenker,. Pricing in computer networks: reshaping the research agenda. *Telecommunications Policy*, (Special issue: Lessons from the Internet), 20(3), April 1996.

[Scherer88] Joan, R. Scherer, Thomas A. Murray. Management of the Intelligent Network. *IEEE Communications Magazine*, 26(12): 21-24, December 1988.

[Schnizlein98] John M. Schnizlein. How Can Routers Help Internet Economics?. *In Proceedings ICE'98* Charleston SC, USA. Pages 52-58. ACM Press, 1998.

[Schulz92] Klaus Schultz, Heinrich Kaufer, Bernhard Heilig. The Deutsche Bundespost Telekom Implements IN. *IEEE Communications Magazine*, 30(2): 90-96, February 1992.

[Senior91] Paul Senior. An Architecture for the Implementation of an Integrated Management System. In I. Krishnan and W. Zimmer, editors, *Integrated Network Management*, pages 359-370. Elsevier Science Publishers B.V., 1991.

[Shapiro99] Carl Shapiro and Hal R. Varian. Information rules: A Strategic Guide to the Network Economy. Harvard Business School Press, Boston Mass.,1999

[Shrewsbury95] J. Kirk Shrewsbury. An Introduction to TMN. *Journal of Network and Systems Management*, 3(1): 13-38, 1995.

[Stanlake95] G.F. Stanlake and S.J. Grant. *Introductory Economics*. Longman 1995.

[Szyperski97] Clements Szyperski. Component software. Addison-Wesley 1997.

[TINAB99] Hendrik Berndt et al. The TINA Book. Prentice Hall Europe, 1999.

[TINABM97] Harm Mulder (ed.). TINA Business Model and Reference Points. TINA Consortium May 1997

[TINAGOT97] TINA Glossary of Terms. TINA Consortium 1997.

[TMFBoM98] Telecommunications Operations Map. TeleManagement Forum.

[TMN00] Keith J. Willetts and Elizabeth K. Adams. Evolving TMN as Global Telecommunications Prepares for the Next Millennium. IEICE Transcations on Communications, e80(B.6), June 1997.
Available at http://search.ieice.or.jp/1997/pdf/e80-b_6_796.pdf

[TNRA97] Frank Steegmans (ed). TINA Network Resource Architecture. TINA Consortium, February 1997.

[TOCP95] Martin Chapman (ed.). Overall Principles and Concepts of TINA. TINA Consortium Feb. 1995.

[TReq95] Requirements upon TINA-C Architecture (document label TB_MH.002_2.0_94). Stefano Montesi (ed.). TINA Consortium, February 1995.

[TSA95] Hendrik Berndt and Roberto Minerva (eds). TINA Service Architecture Version 2.0, TINA Consortium March 1995.

[TSA97] Lill Kristiansen (ed.). TINA Service Architecture Version 5.0. TINA Consortium, June 1997.

[TSCS97] Per Fly Hansen and Carlo Alberto Liccardi. TINA Service Component Specification. TINA Consortium January 1998.

[TSDG95] Johan Sallros (ed.). TINA-C Service Design Guidelines. TINA Consortium March 1995.

[TSMWG99] TINA Service Management Working Group. Available at

[Tran96] Frédéric Dang Tran, Victor Perebaskine and Jean-Bernard Stefani. Binding and Streams: the ReTINA approach. In *Proceedings TINA'96 Conference* (ISBN 3-8007-2201-1), VDE-VERLAG, Berlin & Offenbach, 1996.

[Tt99] TINA Tutorial Slides.TINA Consortium 1999. Available at: http://www.tinac.com/tutorial/tutorial.htm

[VD1197] ACTS Project Vital (AC003). Deliverable D1: Description of the Second Trial (Part A) Available at http://www.algo.com.gr/acts/dolphin/cdrom/html/vital/deliverables/d11/d11b1aw6/D11b1aw6.html

[VitalA298] ACTS Project Vital (AC003) Profile. Available at http://www.algo.com.gr/acts/dolphin/cdrom/html/vital/profile/vital.htm

[Wade00] Vincent P. Wade and Tony Richardson. Workflow – A Unifying Technology for Operational Support Systems. In *Proceedings of 2000 IEEE/IFIP Network Operations and Management Symposium (NOMS 2000)*. IEEE Computer Society Press 2000.

[Wang96] Qiong Wang, Jon M. Peha and Marvin A. Sirbu. Optimal Pricing for Integrated Services Networks. In Lee W. McKnight and Joseph P. Bailey editors, *Internet Economics*, pages 353-376. The MIT Press, 1997.

[Weisser88] Frank J. Weisser and Randall L. Corn. The Intelligent Network and Forward-Looking Technology. *IEEE Communications Magazine*, 26(12): 64-70, December 1988.

[Wilson93] Robert Wilson. Nonlinear Pricing. Oxford University Press, 1993.

[X.74298] X.742. Usage Metering Function for Accounting Purposes. Draft international standard ISO/IEC 10164-10 / ITU-T Recommendation X.742. 1998.

[Yordon84] Wesley J. Yordon. Telephone Rates: Economic Theory and Current Issues. In Leonard Lewin, editor, *Telecommunications: An Interdisciplinary Text (ISBN 0-89006-140-8)*, chapter 6, pages 237-271, 1984.

[YoungJ88] John W. Young. Service Creation: A Telephone Company Perspective. *IEEE Communications Magazine*, 26(12): 64-70, December 1988.

[YoungJS99] Jeffrey S. Young, ‘The next net’, *Wired Magazine*, April 1999.

[Zave93] Pamela Zave. Feature Interactions and Formal Specifications in Telecommunications. *IEEE Computer*, 26(8):20-29, August 1993.

Appendix A: Published Papers

Appendix B: Accounting Component IDL Interfaces