THE WANDERING LOGIC OF INTELLIGENCE
Or Yet Another View on Nomadic Communications

Plamen L. Simeonov
Siemens AG – Berlin, Information and Communication Networks, Communications on Air, CTO

Key words: Intelligent Networks, Service Architecture, Network Convergence, Nomadic Service Logic.

Whatever is flexible and flowing will tend to grow ... (Dao De Jing)

Abstract: This paper presents a third generation (UMTS/IMT2000) integrated network architecture among Flexible Intelligent Network Elements (FINEs) based on the multimedia multimode (client, server and agent) IN Service Node concept ([1],[2]), to perform a crafty Unified Media Communications Service (UMCS) across heterogeneous networks. The FINE architecture represents a network of configurable elements allowing user access to a Unified Message Store (UMS) and Universal Communications Channel (UCC) via traditional PSTN/ISDN/PLMN equipment such as telephones, pagers and fax machines on the one side and networked computers and mobile terminals equipped with mail readers, web browsers and video codecs to enable both on-line and off-line interactions. A virtual cluster of FINEs within an integrated personal communications network is dynamically configured in a distributed or centralised manner according to user profile requirements, network size and performance to provide artful messaging and conferencing services in a changing environment. Each FINEC can be deployed in some of the three modes --- independent (server), dependent (client) and autonomous (agent) --- with respect to the user/network configuration, thus allowing a dynamically configurable (per user/ per node/ per service) centralised or distributed service architecture. The FINE itself is organised along with other nodes by a Network Operation Support Environment (NOSE), a service oriented and TMN compliant Operation, Administration & Maintenance Center. Service logic can be transferred, installed and mounted on demand among the FINE Controllers and the terminal equipment to provide optimal QoS.

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

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

1. INTRODUCTION

The core of the proposed new IN architecture is the *FINE*, the result of a joint project initiative, *@INGate*, between Siemens AG and the Technical University Berlin, [1]. The goal of this project was to design an InterWorking Unit (IWU) as a mediation device between existing circuit and packet switching networks by using traditional ISDN/PSTN/PLMN equipment (TE2) and Internet Terminals (IT), cf. Fig. 1.

Our prototype hosts a generic Unified Media Communication Service (UMCS) composed of multimedia messaging (store-and-forward), telephony and agent-based network computing.

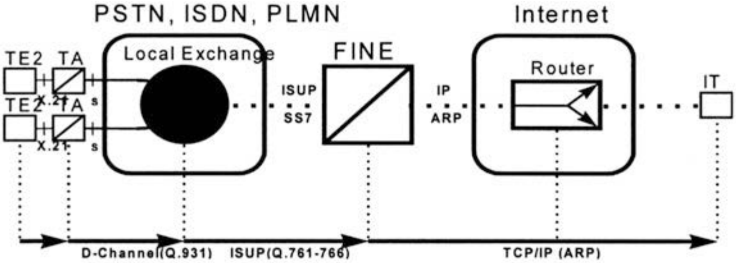

Figure 1. The Network-Bridge FINE as an InterWorking Unit

2. THE REFERENCE MODEL

Our Reference Model contains the following network elements (Figure 2):

- Flexible Intelligent Network Element (FINE)
- FINE Data Server (FineDS)
- Network Operation Support Environment (NOSE)
- WWW Server and Firewall

The Network-Bridge FINE architecture represents an InterWorking Unit between PSTN/ ISDN/ PLMN and the Internet. It is based on the Service Node concept in Intelligent Networks that was extended by cunning media treatment and routing techniques along with enhanced SRF, FSL and scheduling mechanisms.

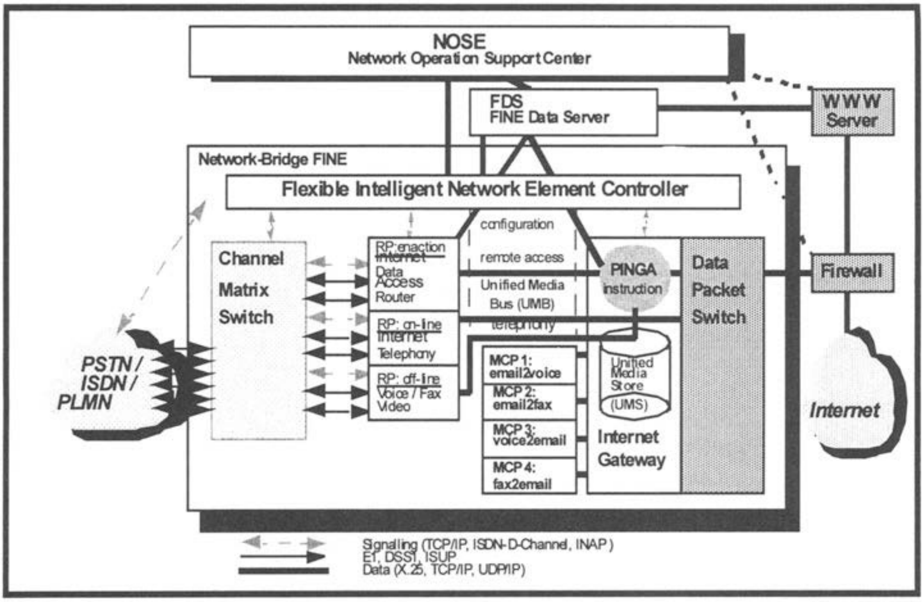

Figure 2. The @INGate Reference Model

The FINE consists of a FINE Controller (FINEC), one or more Resource Platforms (RPs), a Channel Matrix Switch (CMS), a Data Packet Switch (DPS), Media Conversion Processors (MCPs) and an Internet Gateway (IG) with a Unified Media Store/Bus (UMS/B), and a Personal IN Gateway Area (PINGA).

All FINE components except the FINE Controller which carries multiple service logic are optional and configurable to allow a *multimode* functionality of each single element.

The FINE Data Server (FineDS) holds all service and subscriber relevant information such as service configuration and user profile records.

The Network Operation Support Environment allows access to all FINE components (and of the neighbor FINEs) via remote control sessions to UNIX/WWW hosts (FINECs) or via SNMP-based management.

The *Firewall* computer connects the FINE to the Internet.

The WWW Server supports WWW-based management of all FINE components in addition to the TMN-conform NOSE workstation, as well as Customer Service Control (CSC) such as configuration of service specific parameters through the Internet by using security-enhanced Web browsers. Please refer to [2] for details about the model.

3. THE COLLISION OF THE NETWORK PARADIGMS

There is a definite distribution of functions and roles among network components within outbound-signaling telephony based IN. Thus, the standard SCP-IP relationship is e.g. not simply the client-server one (typical for data networks), but rather a processor-coprocessor delegation of service logic (e.g. playing announcements and collecting user input) to a subordinate unit, the Intelligent Peripheral, which delivers back some result to the Service Control Point. The SCP then continues to control the flow of the service in the well-known *main()* routine manner. Even in a distributed IN, Service Nodes are not supposed to serve each other in the common ''data packet'' sense, but rather to have a local (centralized) control over certain resources, services and subscribers. In addition, ongoing IN standardization do not consider alternative network technology integration paradigms, but rather extend the SS7 model by ROSE functionality, such as e.g. invoking functions (cf. ETSI INAP CS-2).

On the other hand, the packet data world is handling intelligence not in a call flow but rather in an application oriented way while addressing *variable* bandwidth technology, resource reservation (RSVP, [15]) and OSI protocol *stack permutations* such as e.g. running SS7 over TCP/IP ([6]).

Such developments are a clear sign that the layered architectures and the *fixed* treatment of IN functions, databases and services can no longer exist in a heterogeneous distributed network under rapid development where even *sculptured object paradigms* such as CORBA, TINA, or DCOM and powerful IN architectures based on ''clustered, multiprotocol SCPs" (Figure 3) cannot handle all aspects of the network evolution as far as they remain *rigid.* One major aspect of the network has been neglected until now: it is *temporal* and *living,* i.e. *changing.* For example proprietary IPs can grow up to autonomous multiprotocol ''virus-like'' CTI point solutions hiding micro data networks in themselves.

A time for change in network design thinking has come. Now, we have the chance to adopt a *stepwise approach* for building the kernel of a new communications cell. Where could we start from? Perhaps *flexible nomadic intelligence* may be an option.

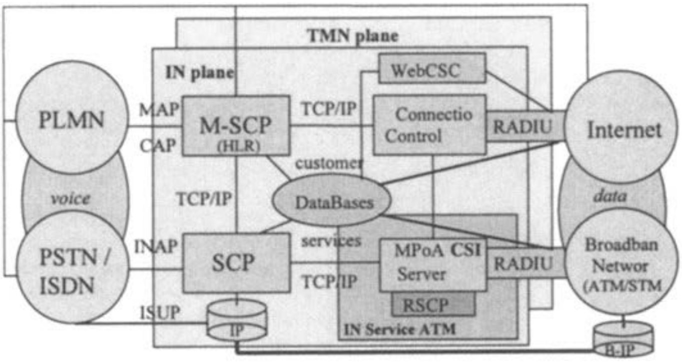

Figure 3. A Clustered SCP Configuration for Converged Networks

Solution Approach: The FINE Controller is the heart of the FINE node. It handles the allocation and deallocation of resources and the overall local logic of the service. The basic concept behind a *vivid nomadic network intelligence* is that each FINEC can be deployed in some of the three modes --- independent (*server),* dependent (client) and autonomous (agent) --- with respect to the user/network configuration.

This model allows a dynamically centralised or distributed service architecture. Services are implemented in the FINEC by *true* Flexible Service Logic (FSL) components interfacing to their subparts, micro service logic (μSL, μ-services), and resources in the resource platforms (RP) directly or via interfaces to other FINEs in the network according to the 'multiple SCP-IP' model, cf. Figure 4.

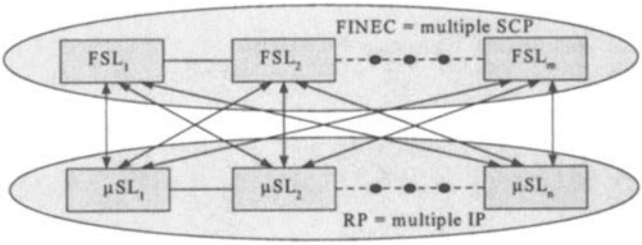

Figure 4. Distribution of Service Logic between SCP and IP

In the following, I will show how flexible intelligent network elements can be used for functionally integrated multimode logic control in advanced *nomadic intelligence* architectures.

4. THE FINE ARCHITECTURE

Figure 5 below illustrates the idea of a configurable network node that is essential for the FINE architecture.

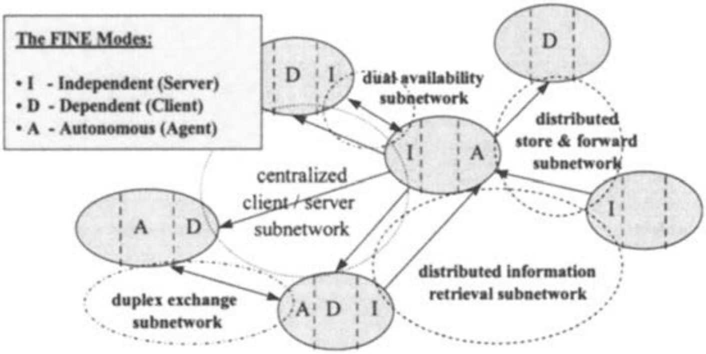

Figure 5. An Example of a FINE Configuration

Definition 1: A *Flexible Intelligent Network Element (FINE)* is a configurable multimode network element. It can be deployed in a single, dual or triple mode of operation as independent (I), dependent (D) and autonomous (A) unit, i.e. as *server*, *client* or *agent* accordingly. All three modes can be deployed at the same time as far as this does not conflict with basic network design principles. A FINE may be a physical entity, a piece of software or a virtual subnetwork by itself.

Even network terminals can be regarded as FINEs operating in some of the three modes.

Definition 2: *The FINE Architecture* is a versatile network intelligence construction of FINEs which are *temporarily* adopting certain roles in the network. It is required that

- agents be assigned to negotiate about the distribution of roles among FINEs (active network), and
- data, functions (service logic) and roles (modes) migrate upon request from one FINE node to another, thus allowing for dynamic network configuration.

This simple model offers a unified structured and evolutionary approach to intelligent network design and configuration which addresses the two basic architectures, centralized and distributed IN and allows for a flexible, but *deterministic* (event-driven) *spread of intelligence* across the network matching user demands wherever and whenever required.

This approach allows for effective treatment of problems such as feature interaction, service mobility (CAMEL+), database updates, etc.

The FINE model can be easily implemented using approaches such as JTAPI [4] for applications, JINI [14] for services and devices, or ATP (Agent Transfer Protocol) [5] for transferring service logic as mobile agents between networked computers.

The main advantage of this approach is that it allows sophisticated network growth, adaptation and rapid introduction of new services by making only minor changes on the available infrastructure. In this way, the FINE architecture represents a *vivid object-oriented model* for intelligent service provisioning and control when compared to the traditional horizontal and vertical OSI-like layered IN architecture.

4.1 Example A: Centralized FINE Architecture

Let us have a centralised loop video-on-demand service architecture (cf. [16], Figure 6) where media contents are transferred from a Central Media Server (CMS) to terminals throughout the network. Signalling and control are performed centrally (inbound or outbound) using the SCP-IP dialogue scheme. Media transfer is performed in a continuous synchronous channel mode or in a discontinuous asynchronous buffering mode. The CMS system provides media transfer, control and management from a single site. If the system fails or becomes incapable to support connections, connections are dropped unless there is a backup CMS.

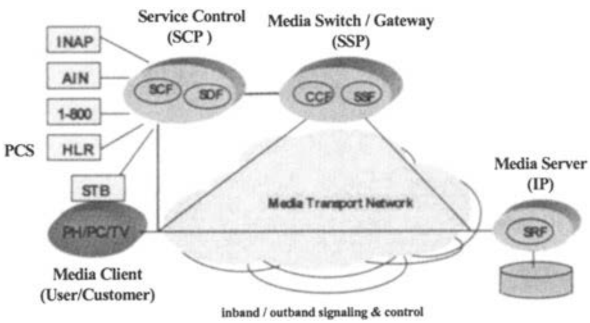

Figure 6. Centralized FINE Architecture

Here the FINE architecture can be configured to represent a centralised IN architecture. It requires three network elements: a media server (I mode), a media terminal client (D mode) and a control point server/client (I/D modes).

A high availability architecture as a second SCP and additional CMS can be modelled by introducing two server/client/agent (I/D/A modes) and two server/agent (I/A modes) elements instead of the single server/client and the single server respectively.

4.2 Example B: Distributed FINE Architecture

Let us now have the same video-on-demand service as a distributed multi-server architecture that spreads out the CMS functionality throughout the network using the concept of Local Media Server (LMS). This configuration saves communication costs and increases system reliability at the expense of local storage capacity by allowing subscribers to access media directly via Local Distribution Switches (LDS).

The central servers are located at the Media Information Provider (MIP) sites (Figure 7). Each LMS may be a mini-CMS and its media contents may be downloaded off-line from some CMS and updated periodically.

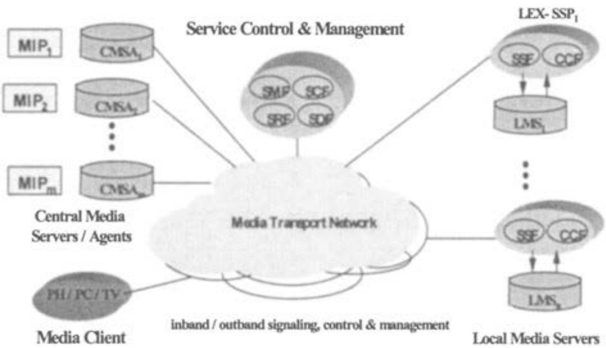

Figure 7. Distributed FINE Architecture

Here FINE may be configured in several variants from a cluster of purely centralized subnetworks of *m* CMS sharing *n* LMS to a cluster of purely distributed subnetworks of *n* LMS. In a distributed subnetwork, an LMS can be represented as a server (I mode) and a media terminal as a client (D mode). Modeling high availability is similar to the centralized case.

Furthermore, a single FINE may become an agent (A mode) to match SCP/SMP functionality by identifying LMS locations within distributed subnetworks and managing the overall service within the macro-network.

Of course, this function can be also cloned into a cluster of managing agents, one for each subnetwork. Then, why should not we keep this one and other options free for choice and configure the *network on demand*?

5. NOMADIC SERVICES AND THEIR LOGIC

The idea of migrating service functionality comes out of traditional network convergence solutions such as the One Number Service (ONS) for fixed and mobile networks. Figures 8 and 9 illustrate two possible variants to realize this service architecture.

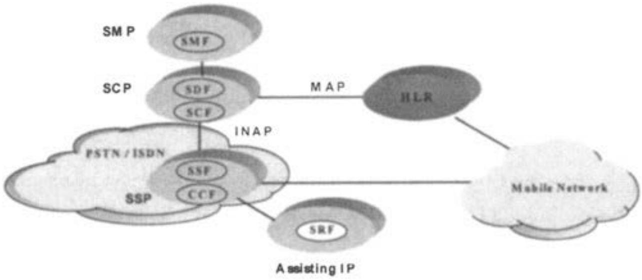

Figure 8. One Number Service for PLMN and PSTN based on Assisting IP§

The essential difference between them is the location of the Service Resource Function (SRF). We can simply ask what would come out to being if we *let this function roam* between the network elements for the reasons of evolution, effective resource utilization, adaptability and QoS guarantee.

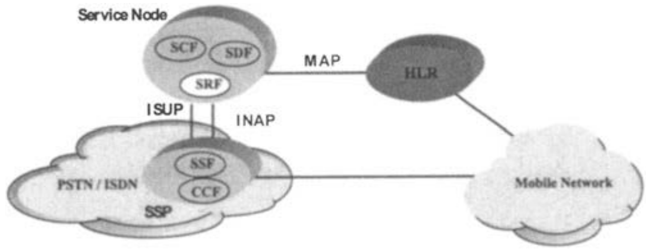

Figure 9. One Number Service based on Service Node with Integrated SRF §

To assist new services and customers, intelligent networks require permanent upgrades in hardware and software components which are not always optimally exploited at their location (Figure 10). Even a distributed architecture which dedicates certain resources for local usage is not adapting (not yet) to variable load and service usage.

§ - with the kindly permission of Dr. R. Rieken, Siemens AG: IN'97 Workshop (IEEE), May 1997, Colorado, USA·

SCF + SDF : Number Translation, VPN, ...

SCP + SDP + SRF : Voice Recognition, UMS, ...

SCP + SDP + SRF + SSF : Local Operators, ...

SCP + SDP + SRF + SSF + GWF : Internet Telephony ...

Figure 10. Overall Trend - Increasing Enhancement of IN Functions

Thus, next to *HOW*, the question of *WHERE* to physically implement an IN function has always been crucial for network designers to provide for more efficient and effective services. Now, in the era of platform independent OO technology this subject obtains a new thread: *mobility*.

Therefore, we still have two choices:

- a non-deterministic worst-case-dimensioned service within a rigid network architecture requiring evolutionary software and hardware upgrades, or
- a deterministic service within a dynamically configurable network architecture, *FINE,* based on autonomous nomadic services taking care for software updates and optimal utilization of hardware resources to a certain grade that can be proved by algorithms.

In some cases, it seems reasonable to temporarily enact and allocate/move different services and their resource areas depending on the eventual customer demands, instead of permanently increasing the functionality and the capacity of the network elements (Figure 11).

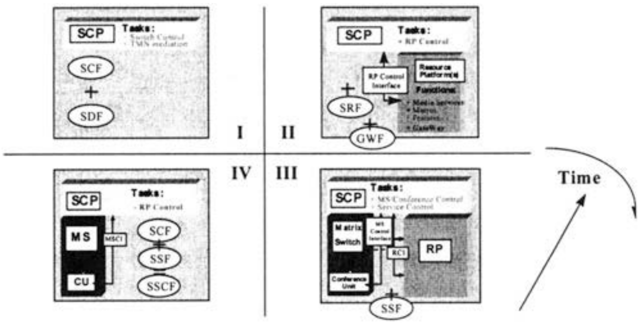

Figure 11. A Possible FINE Node Utilization Cycle

This implies an event-driven, i.e. deterministic, redistribution of roles among the network elements (i.p., client, server and agent) to satisfy network performance and user demands on configuration and QoS (cf. Figure 12).

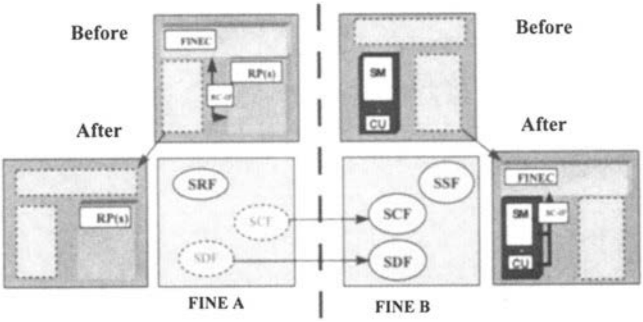

Figure 12. A FINE Reconfiguration via Function Split, Migration and Clustering

Definition 3: *Nomadic FSLs* (flexible service logic) are the smallest autonomous service elements known as features (functions) including parameters (variables) and database contents being transferable from one (fixed or mobile) node of the network to another to offer *intelligence on demand.*

Thus, the FINE model leads to an adaptable, learning, self-configurable service intelligence architecture where the database and feature mobility are parts of the service itself. Although the FINE approach has an IN telephony origin, it can be used to virtually (re-)configure in a dynamic manner *any type of network.*

6. THE WANDERING LOGIC OF INTELLIGENCE

Recent research on active networks approaches the problem of effective service provisioning not by moving services along network nodes, but rather by using the means of (mainly Java driven) object-oriented encapsulated software technology up to the deepest layers and elements of network protocols. No matter of whether we talk about a *Netscript* to program functions for sophisticated packet stream processing in network nodes [7], evolving to form *capsules* or datagrams carrying code [8], *sprokets* or "smart packets" [9], *switchlets* or "active packets" [10], *CANEs* or "composable active network elements" [11], -- in all these works the search for data and process customisation beyond the limits of performance, up to the application, up to the user needs is evident.

The belief creates the reality. This holds until the next change of the paradigm. Yet, we still miss the quantum jump, the profound uncertainty equation that tells us where are the limits of networking management.

For the time being, there are basically two approaches to improve an application's performance: a service oriented and a network oriented, both considered as *objects*, both strictly separated in hardware and software.

Programming is mainly a *soft* method that leaves certain options and degrees of freedom to adjust a model's behaviour towards a desired result. If a high-level abstraction program performs well, it is then translated into a "lower" language and eventually split into modules and instructions that fit the hardware and thus better perform. Finally, if this works well, the entire program is "burned" into silicon to match the "click-to-switch" or "be there" requirement and perform for the best. Further improvements are only due to the physics.

Unfortunately, there is no way back ... to check if the things could work the other way up.

If you have a new idea, you need a new model that most probably needs a new language to express its axioms, predicates etc. logic constructs. The trouble with modern computer science, however, is that it still lacks the freedom of the greek philosophy era. Every problem is digitalized and reduced to a set of objects and functions (both *corpuscules*) that express the same formalism in the broad sense. Be in hardware or software, it is the same logic and the same *programming* language, no matter whether we unfold services or inject pro-active capsules into network nodes.

What we probably miss is the analogous *wave* in the digital world, the sparking *process* that describes the quantum nature of evolving multiple realities, some kind of a free, *Wandering Logic of Intelligence (WLI)* that can be at least welcome by the Lamport's strong fairness formula [13]:

$$SF\,(WLI) = (\Box\Diamond\,\langle WLI\rangle_f) \vee (\,\Diamond\,\Box\neg Enabled\,\langle WLI\rangle_f)$$

where $\langle WLI\rangle_f = WLI \wedge (f' \neq f)$, and f is its state function.

7. CONCLUSIONS

The FINE architecture presents a new service view at IN (in ODP terms) when compared to the classical vertical and horizontal layering of intelligence in networks. It can be centralized, distributed or both at the same time depending on the network size, architecture, service configuration, performance, scheduling and QoS requirements, as well as on the subscriber and service migration flow.

This paradigm can be used as a *unified, vivid object-oriented IN* model where the roles of the network elements (SSP, SCP, SMP, SN, IP, servers, routers, CPE, etc.) are temporary and on demand.

Within the FINE concept of a dynamically configurable intelligent network, services and their logic are gaining a new value by becoming *autonomous and truly mobile entities* throughout the nodes of the network reaching even the terminals to turn them temporarily into agents, clients or servers whenever and whenever required. Thus, service logic can be transferred, installed and mounted on demand among the FINE Controllers and the terminal equipment to provide optimal resource utilisation and QoS.

The FINE Architecture Advantages

Replacing the Deterministic with
1. **Undeterministic Approach to Service Deployment**
   according to Actual User Requests and QoS Requirements

Replacing the Fixed Distribution of Roles within the IN and Data Networks
incl. Tunneling (Encapsulation) of Communications with
2. **Flexible Distribution of Roles and Functions (SCF, SDF, SRF, GWF, etc.)**
3. **Dynamic Network (Re-)Configuration**on User Request involving

- *Temporary Load Distribution*
- *Temporary Resource Sharing, Control & Distribution*
- *Variable Channel and Bandwidth Allocation*
- *Reuse of Software and Equipment*

The FINE architecture is still under discussion and refinement. Currently, Siemens is offering a Messaging Gateway as a front-end solution to the SCP (INXpress 6.1) for multimedia services at the boundary to Internet. Yet, this function *may move* to other network elements in future.

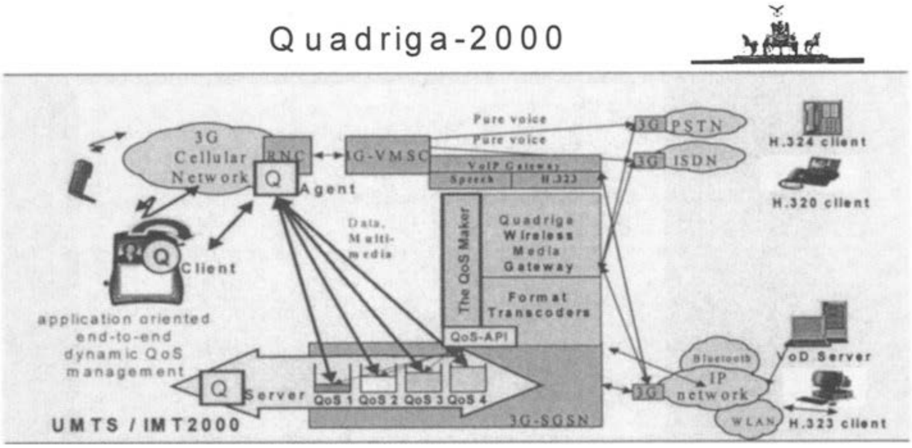

Figure 13. *The QoS Maker Management Architecture for 3G Wireless Video Telephony and VoD over IP*

A typical example for applying the nomadic service logic *approach* is the *dynamic QoS management* (Figure 13).

Siemens is currently investigating this subject in an active network architecture for 3G *wireless* multimedia delivery with the Quadriga-2000 initiative. For more details about this project please refer to the author.

8. ACKNOWLEDGEMENT

The author would like to thank Prof. Dr. Dietrich Reschke from the Technical University of Ilmenau and Dr. Andrea Heilingbrunner from Siemens AG, Munich for the support that made this work appear.

REFERENCES

[1] Plamen L. Simeonov, *Add-On IN Gateway Services for Service Node in Distributed Telecom Networks*, Nr. P30308-A8720-A000-04-7618, Siemens AG, Oct. 1996.

[2] Plamen L. Simeonov et all, *A Smart Node Architecture Linking Telephony with the Internet*, Proc. of ICCC'97, Cannes, Nov. 18-21, 1997, 77-81.

[3] Tsong-Ho Wu et all., *Distributed Interactive Video System Design and Analysis*, IEEE Comm. Mag., March 1997, 100-108.

[4] http://java.sun.com/products/jtapi/index.html.

[5] http://www.trl.ibm.co.jp/aglets/atp/atp.htm.

[6] http://www.ttiweb.com/corporate/press/ssip-pr.htm

[7] Y. Yemini and S. daSilva, *Towards Programmable Netweorks*, IFIP/IEEE Intnl. Workshop on Distributed Systems: Operation and management, Oct. 1996.

[8] D. Tennenhouse et al., *A Survey of Active Network Research*, IEEE Commun. Mag., Jan. 1997.

[9] B. Swartz et al., *Smart Packets for Active Networks*, http://www.net-tech.bbn.com/smtpkts/smart.ps.gz, Jan. 1998.

[10] D. Alexander et al., The SwitchWare Active Network Architecture, IEEE Network, Special Issue on Active and Programmable Networks, May/June 1998.

[11] E. Zegura, *CANEs: Composable Active Network Elements*, Georgia Tech., http://www.cc.gatech.edu/projects/canes/.

[12] D. S. Decasper et all., *A Scalable High-Performance Active Network Node*, IEEE Network, Jan/Feb 1999, 8-19.

[13] L. Lamport, The Temporal Logic of Actions. SRC Res. Report 79, DEC, Sys. Res. Center, Dec. 1991.

[14] Jini Connection Technology Specification 1.1, June 1999, http://www.sun.com/jini/specs/jini1_1spec.html

[15] Zhang, L., Deering, S., Estrin, D., Shenker, S., and Zappala, D., *RSVP: A New Resource ReSerVation Protocol*, IEEE Network, September 1993.

[16] Tsong-Ho Wu et all., *Distributed Interactive Video System Design and Analysis*, IEEE Comm. Mag., March 1997, 100-108.