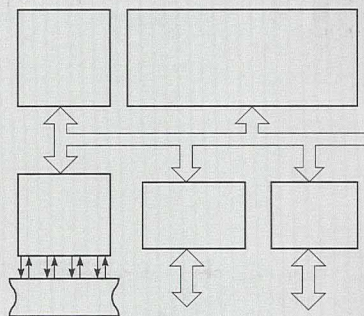
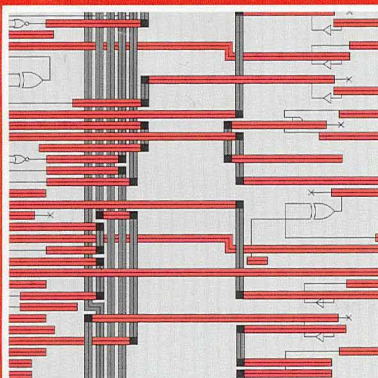
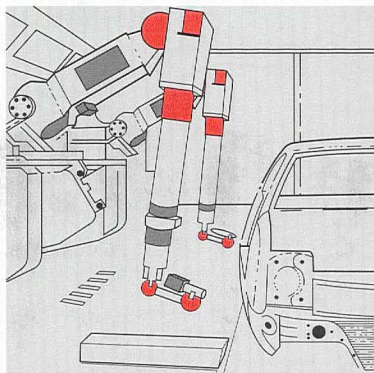


MICROPROCESSORS AND MICROSYSTEMS



Special Issue

Selected papers from
ICCCN 99



Elsevier

Keep track of recently published papers
via the journal homepage at
WWW:<http://www.elsevier.nl/locate/micpro>

MICROPROCESSORS AND MICROSYSTEMS
Volume 24, Number 6, 15 October 2000

Abstracted/indexed in: *Cambridge Sci. Abstr. and Computer Abstracts, Current Contents, Current Technology Index, Electronics and Communications Abstr., INSPEC, Science Citation Index and SciSearch*

Special Issue

Selected papers from ICCCN 99

Contents

<i>A. Sefidcon, F. Khendek</i> FID: feature interaction detection tool	283
<i>P. L. Simeonov</i> On using nomadic services for distributed intelligence	291
<i>A. Calvagna, A. Puliafito, L. Vita</i> Design and implementation of a low-cost/high-performance Video on Demand server	299
<i>F. Chen, B. Li</i> Replicated servers allocation for multiple information sources in a distributed environment	307
<i>S. Vutukury, J. J. Garcia-Luna-Aceves</i> MPATH: a loop-free multipath routing algorithm	319

Keep track of recently published papers via the journal homepage at
[WWW:http://www.elsevier.nl/locate/micpro](http://www.elsevier.nl/locate/micpro)

On using nomadic services for distributed intelligence

P.L. Simeonov*

Siemens AG, Siemensdamm 50, D-13623 Berlin, Germany

Abstract

This paper presents a next generation integrated IN architecture among Flexible Intelligent Network Elements (FINE) based on the multimedia multimode (client, server and agent) IN Service Node concept (Proc. ICCE '97, Cannes (1997) 77), to perform a crafty Unified Media Communications Service (UMCS) across heterogeneous networks. The FINE architecture represents a network of configurable interworking 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 and Web browsers on the other side, 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, telephony and I³E services (Intelligent Inter-, Intra- and Extranet) in a changing environment.

The FINE consists of a Channel Matrix Switch (CMS), several Resource Platforms (RP) containing Media Conversion Processors (MCP) and Channel Managers (CM) to perform the media translation and routing in the required interchange formats, an Internet Gateway (IG) to hold the subscribers' mailboxes and provide the internet connectivity, and a FINE Controller (FINEC) to realise the FINE Service Logic (FSL) and manage the FINEs.

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 Centre.

Service logic can be transferred, installed and mounted on demand among the FINE Controllers and the terminal equipment to provide optimal QoS. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Intelligent networks; Service architecture; Network convergence; Nomadic service Logic

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

1. Introduction

The core of the proposed new IN architecture is *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, [2].

Our prototype hosts a generic Unified Media Communication Service (UMCS) composed of multimedia messaging (store-and-forward), telephony and agent-based network computing to provide integrated information services between telephony and data networks.

2. The reference model

The *@INGate* reference model contains the following network elements (Fig. 2).

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

The Network-Bridge FINE architecture represents an

* Tel.: +49-30-386-24346; fax: +49-30-386-31110.

E-mail address: plamen.simeonov@icn.siemens.de (P.L. Simeonov).

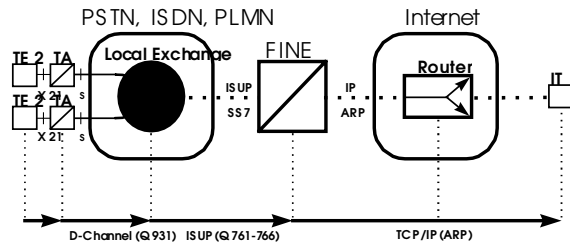


Fig. 1. The Network-Bridge FINE as an InterWorking Unit.

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.

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);
- 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 neighbour FINEs) via remote control sessions to UNIX/WEB hosts (FINECs) or via SNMP-based management.

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

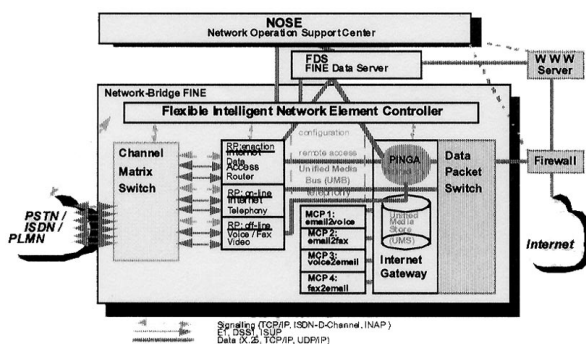


Fig. 2. The @INGate Reference Model.

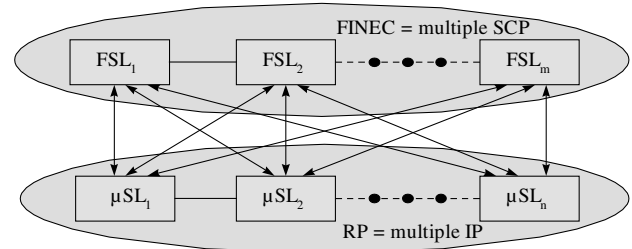


Fig. 3. Distribution of service logic between SCP and IP.

through the internet by using security-enhanced web browsers.

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

3. The FINE architecture

This section reviews the FINE components in detail.

3.1. A FINE Controller

The FINE Controller (FINEC) is the heart of the FINE node. It handles the allocation and deallocation of resources and the overall local logic of the service.

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.

Services are implemented in the FINEC by *true* Flexible Service Logic (FSL) components interfacing to their subparts, micro service logic (μ SL, 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 (Service Control Point-Intelligent Peripheral), cf. Fig. 3.

3.2. Resource Platforms

The Resource Platforms (RPs) are dedicated IPs that implement media delivery and retrieval — reception, storage and response —, as well as Specialized Resource Functions (SRF) such as collecting user input, playing announcements and invoking ROSE operations (e.g. media/user recognition and conversion).

Appropriate RPs are assigned to incoming and outgoing calls, as well as to service logic by the FINEC. There are three different RP types: one for off-line media delivery and retrieval (e.g. unified messaging), one for on-line communication (e.g. internet telephony and conferencing) and one for service enacting via programmable Customer Service Control (CSC) interface — on-line via IVR/VR/DTMF/http/wap and off-line via PING agents.

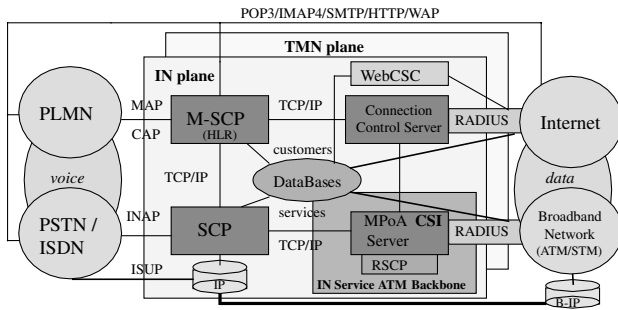


Fig. 4. A clustered SCP configuration for converged networks.

3.3. Channel matrix switch

The Channel Matrix Switch (CMS) is controlled by the FINEC and it routes calls from PSTN/ISDN/PLMN to the appropriate RPs via circuit call related signalling (ISUP, TCAP, MAP).

3.4. Data Packet Switch

The Data Packet Switch (DPS) is controlled by the FINEC and it routes calls from the internet to the appropriate UMS/B or to the PINGA.

3.5. Media conversion processors

The Media Conversion Processors (MCP) have the task of converting media messages upon CSC request:

1. between RP-specific and external formats, and
2. between two different RP-specific formats.

3.6. Internet gateway

The Internet Gateway (IG) basically consists of:

- a Unified Media Bus (UMB) enabling the on-line communication between users in both network domains along dedicated lines for configuration, remote access and telephony;
- a UMS with customer disk partitions for message archiving in a Unified Media Format (UMF); and
- a PINGA for network computing by programming individual agents to fulfil specific tasks such as “wake-up” calls, news delivery or database access. It is connected to the UMS and via the UMB to the resource platforms.

By default, each UMS mailbox functions as an answering machine for the corresponding customer access line (Universal Personal Number, UPN). The PING agents are called, activated and programmed also through the mailbox using a Web/IVR/VR/DTMF interface. They are designed (Service Creation), configured (CSC) and used (Service Deployment) by the customer herself, e.g. by using scripting technology to realise definite information services. The

PINGA paradigm addresses an *active mailbox* in the FINE. In addition, the internet data access router (RP) has a newsgroup agent for threads selected via the CSC menu.

4. The collision of the intelligence paradigms

There is a definite distribution of functions and roles among network components within outbound-signalling 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 SCP. 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 (centralised) control over certain resources, services and subscribers. In addition, ongoing IN standardisation 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 callflow but rather in an application oriented way, while addressing *variable* bandwidth technology 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, multi-protocol SCPs” (Fig. 4) 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.

In the following, flexible intelligent network elements are shown to be useful for functionally integrated multimode logic control in advanced *nomadic intelligence* architectures, which offer completely new perspectives for communication services in converged networks.

5. The FINE architecture

The fine architecture (Fig. 5) represents a self-configurable, adaptable, and domain-aware intelligence architecture,

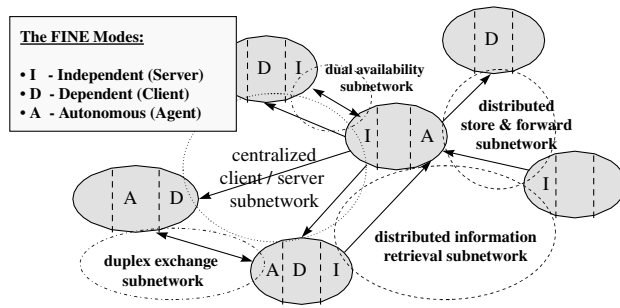


Fig. 5. An example of a FINE configuration.

both in terms of software and hardware, where network properties such as 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 model any type of network.

Definition 1: A FINE (Fig. 5) 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*. 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. It allows for a flexible, but

deterministic (event-driven) *spread of intelligence* across the network, matching user demands wherever and whenever required.

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

The FINE model can be easily implemented using approaches such as JTAPI [4] for applications and 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 in 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.

In the following, I use two examples taken from [3] to illustrate the applicability of the new model.

5.1. Example A: centralised FINE architecture

Let us have a centralised loop video-on-demand service architecture (Fig. 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.

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.

5.2. 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 (Fig. 7).

Each LMS may be a mini-CMS and its media contents

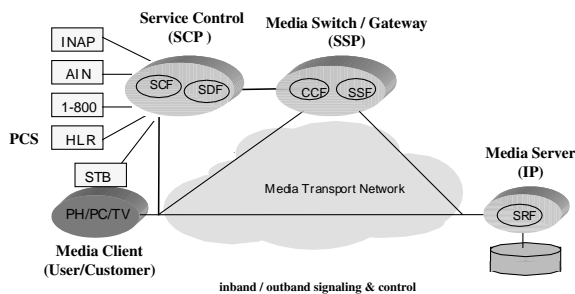


Fig. 6. Centralised FINE architecture.

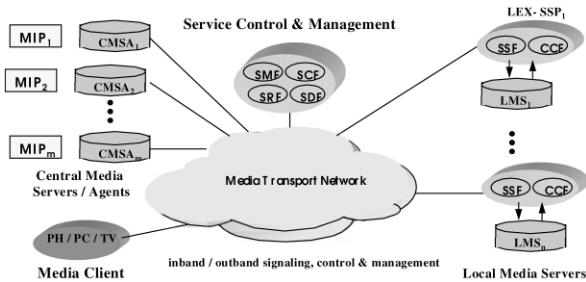


Fig. 7. Distributed FINE architecture.

may be downloaded off-line from some CVS and updated periodically.

Here, FINE may be configured in several variants from a cluster of purely centralised sub-networks of m CMS sharing n LMS to a cluster of purely distributed sub-network of n LMS.

In a distributed sub-network, an LMS can be represented as a server (I mode) and a media terminal as a client (D mode). Modelling high availability is similar to the centralised case.

Furthermore, a single FINE may become an agent (A mode) to match SCP/SMP functionality by identifying LMS locations within distributed sub-networks 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 sub-network. Then, there is no reason for not keeping this one and other options free for choice, and configure the *network on demand*.

6. 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. Figs. 8 and 9 illustrate two possible variants to realise this service architecture. The essential difference between them is the location of the Service Resource Function (SRF). We can simply ask what would happen if we *let this function roam*

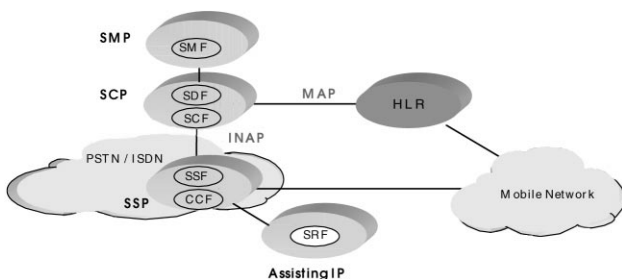


Fig. 8. One number service for PLMN and PSTN based on Assisting IP (with the kindly permission of Dr R. Rieken, Siemens AG: IN'97 Workshop (IEEE) contribution, Colorado, May 1997).

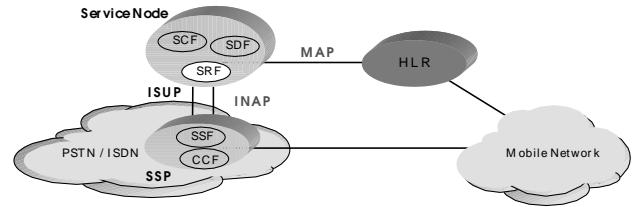


Fig. 9. One number service for PLMN and PSTN based on service node with Integrated SRF (with the kindly permission of Dr R. Rieken, Siemens AG: IN'97 Workshop (IEEE) contribution, Colorado, May 1997).

between the network elements for the reasons of evolution, effective resource utilisation, adaptability and QoS guarantee.

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 (Fig. 10). Even a distributed architecture, which dedicates certain resources for local usage, is not adapting (not yet) to variable load and service usage.

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 utilisation 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 (Fig. 11).

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. Fig. 12).

Definition 3: *Nomadic FSLs* (flexible service logic) are the smallest autonomous service elements known as

- | | | |
|-----------------------------|---|------------------------------|
| SCF + SDF | : | Number Translation, VPN, ... |
| SCP + SDP + SRF | : | Voice Recognition, UMS, ... |
| SCP + SDP + SRF + SSF | : | Local Operators, ... |
| SCP + SDP + SRF + SSF + GWF | : | Internet Telephony ... |

Fig. 10. Overall trend — increasing enhancement of IN functions.

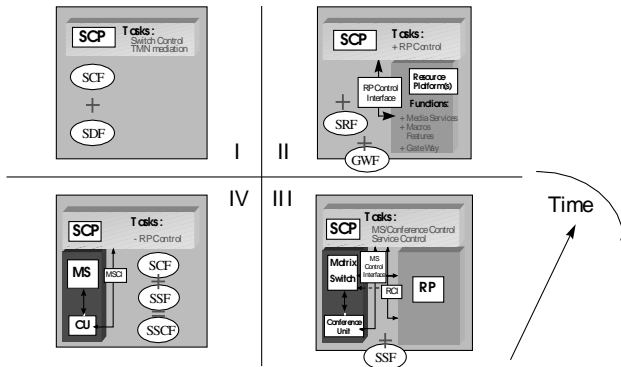


Fig. 11. A possible FINE node utilization cycle.

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*.

7. The wandering logic of intelligence (WLI)

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], *sprockets* or “smart packets” [9], *switchlets* or “active packets” [10], *CANEs* or “composable active network elements” [11], active plugins [12] — 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 so evident, as it was with the definition of different QoS layers matching the OSI-centric communications model a few years ago. The belief creates the reality. This holds until the next change of the paradigm. Yet, we still miss the quantum jump, the profound networking uncertainty equation.

For the time being, there are basically two approaches to improve an application’s performance: service oriented and 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 perform better. Finally, if this works well, the entire program is being moulded 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 digitalised and reduced to a set of objects and functions (both *corpus-cules*) that express the same formalism in the broad sense. Be it 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 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* that can be at least welcome by the Lamport’s strong fairness formula [13]:

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

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

8. 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 centralised, 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 paradigm of a dynamically

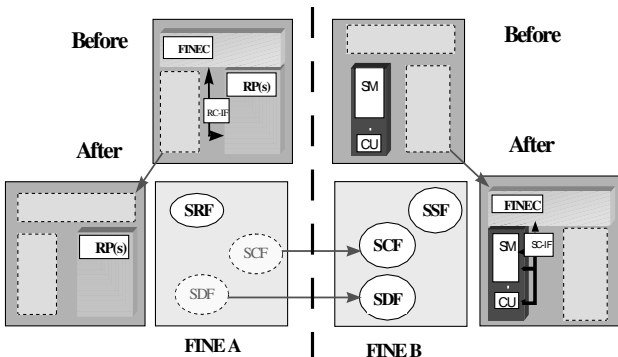


Fig. 12. A FINE reconfiguration via function split, migration and clustering.

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 wherever 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 are as follows. Replacing the deterministic with:

- (1) an 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 including tunnelling (encapsulation) of communications with:

- (2) a flexible distribution of roles and functions (SCF, SDF, SFR, GWF etc.); and a
- (3) Dynamic Network (Re-) Configuration 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 SCP for multimedia services at the boundary to internet (INXpress 6.1). Yet, this function *may eventually move* to other network

elements in future. Whatever has been a network capability in the past may become a system/platform function in the future or vice versa. The choice is open.

References

- [1] P.L. Simeonov, Add-On IN Gateway Services for Service Node in Distributed Telecom Networks, Nr. P30308-A8720-A000-04-7618, Siemens AG, October 1996.
- [2] P.L. Simeonov et al., A Smart, Node Architecture Linking Telephony with the Internet, Proceedings of the ICC'97, Cannes, November 18–21, 1997, pp. 77–81.
- [3] Wu Tsong-Ho, et al., 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, S. daSilva, Towards Programmable Networks, IFIP/IEEE International Workshop on Distributed Systems: Operation and management, October 1996.
- [8] D. Tennenhouse, et al., A survey of active network research, IEEE Commun. Mag. 35, (Jan) 1997 pp. 80–86.
- [9] B. Swartz et al., Smart Packets for Active Networks, <http://www.net-tech.bbn.com/smtpkts/smart.ps.gz>, January 1998.
- [10] D. Alexander, W. Arbaugh, M. Hicks, P. Kakkar, A. Keromytis, J. Moore, C. Gunter, S. Nettles, J. Smith, The switchware active network architecture, IEEE Network, Special Issue on Active and Programmable Networks 12,3 (May/June) (1998) pp. 29–36.
- [11] E. Zegura, CANEs: Composable Active Network Elements, Georgia Tech., <http://www.cc.gatech.edu/projects/canes/>.
- [12] D.S. Decasper, et al., 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, December 1991.