State of the Art of Service Creation Technologies in IP and Mobile Environments

Jorma Jormakka
Helsinki University of Technology
Networking Laboratory
E-mail: jormakka@tct.hut.fi

Henryka Jormakka
Technical Research Centre of Finland
Information Technology
E-mail: henryka.jormakka@vtt.fi

Abstract: The paper gives an overview of service creation technologies for IP, PSTN-IP interworking and for mobile networks. There is a large number of possible future service creation technologies around, such as Parlay, MExE, XML, JAIN, and the present service technologies, like IN and CAMEL are not likely to disappear. The paper classifies the technologies into four groups and discusses the ideas and limitations of technologies for value added services.

Key words: Service creation, IN, CAMEL, OSA, Parlay, TSAS, VHE, JAIN, CPL, MExE, WAP, J2ME, SAT, PINT, SPIRITS, Mobile agents/Active networks.

1. INTRODUCTION

Next Generation Network (NGN) service technologies form a loosely defined collection of technologies for future service creation. Terms like All-IP, Softswitch, Extensible Markup Language (XML), are implied.

Technologies, which remind of the Intelligent Networks (IN), like Customized Applications for Mobile network Enhanced Logic (CAMEL), or Open Service Access (OSA)/Parlay, are also included. Additionally, Universal Mobile Telecommunications System (UMTS) has its role in the new service technologies. In fact, without services UMTS would most likely fail, therefore service technologies are crucial for the concept. UMTS and 4th generation mobile networks contribute to the area: wireless terminals contain service technologies like Mobile Execution Environment (MExE) and SIM Application Toolkit (SAT). ExE includes Wireless Application Protocol (WAP) and Java 2 Micro Edition (J2ME), which are relevant on their own as well. Also the Virtual Home Environment (VHE) concept has an important role in service creation. VHE is an elaboration of the personal mobility idea, first introduced in the IN as the UPT service.

NGN service technologies can be divided into four groups. The first group follows the Intelligent Network concept. Intelligent Network is today the predominant service creation technology in voice services in GSM and PSTN. To this group belong CAMEL - an IN technology which allows roaming GSM users to access IN services, PINT (PSTN and Internet Interworking) and SPIRITS (Services in the PSTN/IN Requesting Internet Services).

The second group of service technologies has borrowed ideas from IN and TINA (Telecommunications Information Networking Architecture), but do not need IN type Service Control Points (SCPs) for accessing services. This set includes Parlay, OSA, TSAS (Telecommunications Service Access and Subscription) and JAIN (Java APIs for Integrated Networks). Usually these technologies are understood as service technologies for IP networks, but they do not assume that the network is IP.

The third group of service technologies contains technologies, which mostly reside in a wireless terminal. WAP and I-Mode were the first technologies of this type. To this group belongs also J2ME, an edition of Java running in small wireless terminals like mobile phones, MExE - an enhancement of WAP and J2ME, and SAT - a related technology, where service logic resides in the SIM or Universal Subscriber Identity Module (USIM) card.

The fourth group is a loosely defined set of additional service technologies. Active networks and Mobile Agent technologies fall into this group.

In the presented paper the most important service creation technologies and their current state are described. The conclusions discuss the limitations and future of service creation.

2. SERVICE CREATION TODAY

Currently service creation in the circuit-switched side and in IP networks is based on different technologies.

- *Service creation in PSTN/IN:* The services consist of supplementary services of ISDN made as additions to the protocol, and of IN services. New services are created as IN services using Service Creation Environments (SCEs), which enable definition of services as Service Independent Building block (SIB) chains. Trigger tables must be filled for giving triggering rules for Trigger Detection Points (TDPs), but for creation of new simple services this is enough. Totally new service types require implementation of new Capability Sets (CSs) of IN. Currently only IN CS1 and parts of IN CS2 are in commercial use.
- *Service creation in the Internet:* Basic Internet data services, telnet, ftp, smtp, have been created as data communication protocols on the application layer. Protocol development meant writing Unix-style programs on top of the socket interface, as no specific service creation methods have been used. Only after the invention of HTTP, there exists now a service creation method: WWW services are created using HTML, Java applets, Common Gateway Interface (CGI), or e.g. Perl scripts.
- *Service creation in 2G:* The early way of creating services was adding GSM supplementary services to the protocol. Many new services can be created using these inbuilt features. IN technology CAMEL is needed for making services that roaming users can use, but CAMEL is also seen as a flexible way to create new services. GSM-based 2G services include: Voice calls with GSM supplementary services, group calls etc., Packet data call, Short Message Service (SMS), WAP services: games, entertainment, ticket booking etc. On top of these services is the vendor specific IN platform with CAMEL.

As a summary, in all the three areas service creation has been separated from protocol development. There is a need for a service platform and a service creation process by which new services can be created fast. A natural short-term goal has been to integrate the good sides of service creation methods enabling PSTN/IN type services in IP networks and IP type services in the PSTN/mobile network side. A long-term goal is to create a safe, easy to use, open service creation environment enabling new business opportunities, that is, creation of the service network architecture.

3. FUTURE OF SERVICE CREATION

3.1 IN (Intelligent Networks)

IN [1] service creation follows the IN Conceptual Model (INCM). Service is defined in the service plane (SP) using free form text. A new service should preferably be created using existing service features as otherwise it may be difficult to implement it on a given CS. The purpose of a CS idea is to list targeted services so that it is possible to understand if a given IN platform is sufficient for implementing a new service. In IP service creation, the idea of a capability set has not got enough attention. A probable reason is that giving target services is considered to be similar to defining the services, but that is not the case.

The next step in INCM is to describe the service in the Global Functional Plane (GFP) using SIBs (Service Independent Building blocks). An IN service is a chain of SIBs starting from a Point of Initialization in the Basic Call Process SIB and ending to some Point of Return. In IN CS1 it is not allowed to have parallel SIB chains. In CS2 this is possible and Points of Synchronization are used to synchronize parallel chains.

In the Distributed Functional Plane (DFP) the service is implemented using Information Flows and Functional Entities. In creation of IN services DFP is needed in arming Detection Points. Statically armed Detection Points are called Trigger Detection Points (TDP) while the dynamically charged are called Event Detection Points. Triggering IN service logic occurs in a TDP according to a trigger table. The structure of a trigger table is not standardized, but typically triggering can be based on prefixes or individual numbers. IN dialog with Service Switching Function (SSF) and Service Control Function (SCF) is started at some TDP and the trigger table must be set.

The last plane in INCM is the Physical Plane (PhP). The issues of PhP, which appear in IN service creation, are possible changes to INAP and placement of the Functional Entities. Although INAP is standardized, usually ETSI CORE INAP is used, there are often small changes introduced by operators and a new service may require some additional information to be carried by INAP. Furthermore, the exact signaling sequences can be made in many ways for the same service. Functional Entities naturally are placed in an IN platform to some Physical Entities, but depending on the load, it may be desirable to realize a new service with a different placement of FEs. For instance, one may use a SCP with SCF and Service Data Function (SDF), but SDF may also be placed on a stand-alone Service Data Point. Lastly, there may be some cases when the IN standard is intentionally not followed because of traffic considerations, for instance, in televoting

signaling should go to SCF for each call, but in an implementation signaling may go to SCP only once per several calls.

This shows that service creation in the IN mostly occurs in the GFP using SIBs. Special Service Creation Environments (SCE) with graphical user interfaces are applied. Often the SIBs SCEs have do not directly correspond to the standard SIBs. For instance Siemen's EWSD SCE has much more and more specific SIBs than the ITU-T IN Recommendations mention.

The capability sets of IN in use today are IN CS1 and partially IN CS2. IN CS3 is fully standardized and IN CS4 is on draft stage. IN CS4 is leveraged to future networks. It is the most interesting for service creation in IP and mobile networks since it contains PSTN - IP interworking. IN CS4 defines interfaces to Parlay and to PINT. SPIRITS is not mentioned in ITU-T IN CS4, but this technology fits in quite naturally.

3.2 CAMEL (Customized Applications for Mobile network Enhanced Logic)

CAMEL, specified by ETSI/3GPP [13], is a technology of GSM Phase 2+, which enables roaming GSM users to access IN service in their home operator's environment. CAMEL adds some new functional entities to the GSM network.

gsmSSF: Corresponds to IN Service Switching Function for GSM voice calls. The gsmSSF functionality is located to MSC.

gprsSSF: Like gsmSSF but for SMS. This functionality is located to SGSN.

ipSSF: A new SSF functionality will be introduced in CAMEL for IP Multimedia Core Domain in 2GPP R5. ipSSF will be placed in CPS. CPS has interfaces with GCSN, HLR and gsmSCF.

gsmSC:. Corresponds to SCF and SCP, but can communicate with HLR using AnyTimeInterrogation. The functionality resides in a physical point, also called gsmSCF.

CAMEL uses a modification of IN INAP protocol, called CAP (Camel Application Part). It is a protocol, by which gsmSSF communicates with gsmSCF. Additionally CAMEL contains modifications to GSM MAP: a new operation, AnyTimeInterrogation is added to MAP and gsmSCF can ask HLR information of the roaming user.

There are two state machines in gsmSSF: originating and terminating state machines (O-BCSM, T-BCSM). They resemble IN CS1 BCSM, but are simpler. They contain a restricted number of Detection Points, especially CAMEL Phase 1. In later phases of CAMEL more Detection Points have been added to O-BCSM and T-BCSM.

CAMEL Phases 1 and 2 are used, but they only deal with circuit-switched networks. CAMEL Phase 3 is a new addition. It includes data services for General Packet Radio Service (GPRS), like the GPRS prepaid service.

Service creation in 3G is based on VHE scenario including CAMEL, OSA, and MExE. Most UMTS calls will have CAMEL processing. Prepaid service is the main service of CAMEL. It continues to be the main service also in CAMEL Phase 3, this time for prepaid charging of IP traffic.

3.3 Parlay

Opening IN to 3pty service providers has been too risky because of security and reliability considerations. Founded in 1998 the Parlay Group [6] is claiming to have a solution in the form of Parlay APIs, through which 3pty applications can in a controlled and secure way make use of network functionality and become independent of the underlying network technology. The network under the Parlay APIs can be IP, Quality of Service IP (QoS IP), or a circuit switched network.

The Parlay APIs consists of two categories of interface – Framework Interfaces and Service Interfaces. The Framework Interfaces provide applications with basic mechanisms for making use of the service capabilities in the network. An incoming call to a Parlay service comes through the Parlay Framework that offers capabilities necessary for the Service Interfaces to be secure and manageable. The client applications are using it to conduct the authentication process with the Framework provider, select an instance of network's service, or subscribe to new services. Examples of Framework APIs are Authentication, Discovery or Trust Management APIs. After the authentication process the applications use Discovery API in order to find handles to the services offered by the Parlay. The network services are accessible through Service Interfaces that offer applications access to the network capabilities such as call management or user interaction.

Service creation using Parlay in practice means writing objects using CORBA. Typical languages are Java and C++. These objects call Parlay APIs directly. In this way Parlay service creation is similar to application development, but writing Java is considered relatively easy.

There is a newer approach for using Parlay APIs, so called Parlay/X. In this variant, a service is implemented with XML PDUs on top of HTTP using the SOAP-mapping. The XML documents call scripts in any standard XML method (like XPath/XLib). These scripts call Parlay APIs. This service creation method is similar to CPL, or HTML/CGI service creation.

3.4 OSA (Open Service Accesss)

OSA, the standardization effort of 3GPP [14], specifies architecture that enables applications to make use of network functionality through open standardized interfaces. It resembles the Parlay architecture and can be considered as a specialization of Parlay to mobile networks. OSA consists of three parts:

- Applications* implemented in application servers,
- Framework*, which as in case of Parlay provides applications with basic mechanisms enabling them to use service capabilities in the network. The mechanisms are provided by framework capability features such as Authentication, Discovery, or Service Agreement,
- Service Capability Servers* providing applications with service capability features such as Call Control or User Location

The Framework's and Service Capability Servers' Service capability Features (SCF) are provided through framework and network interfaces respectively.

Currently the standardization bodies of 3GPP and Parlay Group work to achieve a common OSA model and aim to synchronize the release dates. However, although they specify the same functionalities, the terms are still different. The current API is known as OSA Release 5 or Parlay 3.0.

3.5 TSAS (Telecommunications Services Access and Subscription)

Influenced by the TINA business model and service architecture, the OMG Telecommunications Domain Task Force issued in June 2000 document called "Telecommunications Service Access and Subscription" in which it specified interfaces for handling business transactions among three parties: service providers, retailers and consumers. One of the objectives of TSAS was to improve scalability of the TINA specification and remove the redundancy between TINA objects' interfaces using segmentation. Originated from TINA access session, TSAS model provides segmentation of interfaces related to access and subscription.

The document describes how services can be retailed on behalf of service providers, which in turn offer their services to the retailers. It concentrates on specifying consumer-retailer and retailer-service provider interfaces. The interfaces enable users to subscribe, select and access services that are customized according to user preferences. The subscription interfaces and

operations have been reused in Parlay. The TSAS specification is technically aligned with these of the Parlay Group, which means that the service interfaces specified in the Parlay API can be offered using this specification.

3.6 VHE (Virtual Home Environment)

VHE is a concept of the 3G mobile system for personal service environment (PSE) portability across network boundaries and between terminals which means that the user will have the same interface and service environment regardless of location. PSE is a combination of a list of subscribed services, service preferences and terminal interface preferences. It also encompasses the user management of multiple subscriptions, e.g. business and private, multiple terminal types and location preferences. The PSE is defined in terms of one or more User Profiles.

The VHE concept originated from the 3GPP specification [12], which specifies the VHE business model and requirements for a possible service architecture realizing the concept. The work on VHE conducted by 3GPP and other fora is for promoting a service architecture providing an open Application Programming Interface (API) that resides between the application layer and the service component layer. The API offers access to network information and enables services operated by enterprises outside of the network domain to access the network capabilities. The VHE concept can be realized using existing toolkits such as OSA, Parlay, MExE, CAMEL or USAT.

3.7 MExE (Mobile Execution Environment)

MExE [5] is a specification introduced by ETSI and developed by 3GPP enabling provision of standardized execution environment in user equipment (UE). The UE consists of the Mobile Equipment (ME) and SIM/USIM (Universal Subscriber Identity Module). MExE provides the ability to negotiate the UE supported capabilities with a MExE service provider, in this way allowing applications to be developed independently of any UE platform or operator's service platform. The applications can be executed on a remote server, or downloaded to the UE and directly executed there. Two UEs may also be engaged in a MExE service with each other, with the network effectively supplying the "pipe" and not playing a MExE role in the connection.

To handle the variety of possible UE configurations MExE divides them into so called Classmarks. Currently there are defined three Classmarks with more to follow in the future. Classmark 1 terminals are based on WAP, Classmark 2 groups PersonalJava enabled devices, Classmark 3 terminals

are based on the J2ME Connected Limited Device Configuration (J2ME CLDC).

The technologies in MExE, WAP, PersonalJava and J2ME CLDC MIDP, are important without MExE. The main additions of MExE are in the security and access rights model. Due to the support of ETSI and 3GPP, it is likely that MExE will be supported by vendor equipment in the near future.

3.8 WAP (Wireless Application Protocol)

WAP is the WAP Forum's protocol for bringing the Internet to mobile phones. In WAP 2.0, two protocol stacks are supported and they are combined on the Wireless Application Environment (WAE) layer [8], [9]. The WAP 1 stack consists of Wireless Session Protocol (WSP), Wireless Transaction Protocol (WTP), Wireless Transport Layer Security (WTLS) and Wireless Datagram Protocol (WDP), while WAP 2 stack contains modified TCP/IP protocols: WP-HTTP, TCP*, TLS, Wireless IP. WAP works on any bearer, like CSD, SMS, USSD, GPRS and on any network, like GSM, UMTS, PDC-P, iDEN, CDPD.

The WAP Forum has defined an XML based language Wireless Markup Language (WML). XHTML (eXtensible HyperText Markup Language) is another possibility in WAP Version 2.0. XHTML is a version of HTML following the XML standard, including ending tags and defining DTD in the header. WML definition contains so called deck with several WML pages. The deck is similar to HTML front page, but WML is optimized for over the air interface. A deck is moved as whole and pages of the deck will be locally accessible in the mobile phone. Unlike HTML, WML keeps the state, so going to next or previous page is possible in a WML deck, has timers and variables.

WAP services are typically created using a HTTP-server and a WAP simulator. Creating a service means writing a deck of WML pages. It is put to the server and can be accessed by the WAP phone simulator. WMLScript scripts can be added to the WML pages to give some functionality, J2ME MIDlets can add more complex functionality.

Wireless Telephony Application Interface (WTAI) is another way of creating voice services in WAP. It means calling the WTAI API and is mostly intended for authorized WTA servers.

WAP works mostly in the pull-mode, like HTTP. The currently interesting WAP Push of Version 1.2 makes possible new service types.

From performance point of view, WML has optimizations: WAP Binary XML (WBXML, WMLC) and Wireless Bitmap format (WBMP) for images for the Over-The-Air interface. Still, WML is not especially compact. As a language derived from XML, WML carries all tags for elements. This is

useful for writing browsers understanding pages obtained from anywhere in the Web, but transmitting tags is a waste of bandwidth. WAP access to the Web is through WAP proxies and probably a language more compact than WML is feasible.

The version of WAP supported by most mobile phones is WAP Version 1.1., released by the WAP Forum in June 1999. The current WAP 2.0 version was released August 2001. Version 1.2.1 was still supported by only few models at the end of 2001. Despite of the slow market development, WAP has a strong support as 99% of mobile phone vendors are in the WAP Forum. WAP is still a promising future technology with high hopes for WAP on GPRS.

3.9 J2ME (Java 2 Micro Edition)

Java 2 Standard Edition (J2SE) does not fit into small mobile devices, like mobile phones. Several smaller size versions of Java have been created, like Personal Java for set-top boxes and Personal Digital Assistants (PDA). J2ME is a stripped-off version intended to a range of small devices from embedded computers to mobile phones.

J2ME has two configurations: Connected Limited Device Configuration (CLDC) and Connected Device Configuration (CDC). CLDC is intended to mobile phones and small PDAs, while CDC is intended to replace Personal Java in set-top boxes and more powerful PDAs. CDC is much more like J2SE while CLDC has rewritten virtual machine and restricted rewritten libraries. On top of a configuration are profiles. Examples of CDC Profiles are Personal Profile and Foundation Profile, on CLDC there are for instance MIDP (Mobile Information Device Profile) and PDA profile. MIDP is of special interest as it will be used in mobile phones and can enhance WAP. In general, J2ME should not be seen as a replacement of WAP, but as adding new power.

Java MIDlets correspond to J2SE and J2EE (Enterprise Edition) applets. MIDlets are usually packed to Java JAR-files, which are downloaded over the air from a HTTP server using WAP and executed in the User Equipment. J2ME services are developed using simulators in the same way as WAP services. There are special concerns of size and performance when writing MIDlets, and standard libraries of J2SE usually do not exist. Differences in presentation of MIDlets in implementations of different vendors must often be taken into account with small vendor-specific code segments.

MIDP Over-The-Air (OTA) provisioning partially supports service discovery, installation and upgrading, but retrieval of MIDlets is outside the scope of the current MIDP specification. The mobile equipment must

contain a Discovery Application (DA) and using DA new JAD-files can be downloaded from the Web.

Currently, J2ME runs a bit too slowly for good customer satisfaction and it has not yet filled the promise of Java: write once, run anywhere.

3.10 SAT (SIM/USIM Application Toolkit)

The (U)SIM Application Toolkit (SAT/USAT) of 3GPP [11] is designed to provide a standardized execution environment for applications stored on the (U)SIM card. It provides mechanism, which enables the applications to interact with any mobile equipment, which supports the specified mechanism(s) thus ensuring interoperability between a USIM/SIM and the mobile equipment, independent of the respective manufacturers and operators. Additionally, a transport mechanism enabling applications to be downloaded and/or updated is provided. SAT/USAT applications are more limited than MExE applications, but otherwise SAT/USAT can provide nearly the same capabilities as MExE and ensures a larger degree of freedom, because of more secure SIM environment.

SAT/USAT is one interesting element for the provision of VHE. It specifies the interaction between the SIM card, the ME and the network. The standardisation of these interactions contributes to satisfying the portability requirement of the VHE. Indeed, SIM/USIM cards and their applications become portable over several terminals and networks. Additionally, the increased intelligence in the terminal side will cause that users can start managing their services more independently from their operator. This may reduce the need for service provisioning, but also bring provisioning to a new level – inside the mobile terminals.

3.11 PINT (PSTN IP Interworking)

PINT by IETF [2] is a simple way to use IN services from IP networks. The idea of PINT is that an IP host sends a request to a PINT server. The PINT server relays the request to a PSTN network element, typically SCP, but possibly Private Branch Exchange, MSC, or some other element in a circuit-switched network. Finally, the circuit-switched network performs the action. Though PINT uses SIP, it is not connected with IP telephony. Let us give a sample PINT service: an IP host requests using the Internet that an IN SCP causes a fax to be sent to a fax machine using PSTN (Request to Fax Content). PINT is easy to implement, but it is often considered as a transitional solution as Parlay makes similar things without the IN SCP.

The PINT protocol is developed from Session Initialization Protocol (SIP) and Service Description Protocol (SDP). SIP can be considered

known. SDP is a protocol for multimedia session description, especially for joining in multimedia sessions.

Initially, the following PINT services are defined:

- Request to Call
- Request to Fax Content
- Request to Speak/Send/Play Content

3.12 SPIRITS (Services in the PSTN/IP Requesting Internet Services)

SPIRITS [3] is a new SIP-based protocol by IETF, currently on draft stage. SIP here is not used for establishment of an IP call, and in fact, like PINT, SPIRITS does not have anything to do with IP telephony. SIP is here used as a general purpose signaling protocol and the goal of SPIRITS is to allow PSTN users to access through SCP services in the IP side, and to allow IP users to access IN-based services in the PSTN.

Some services can be made with either SPIRITS or PINT: for instance click-to-dial from a Web page can be a PINT service, or it can be a SPIRITS service. It is also possible to make this service using Parlay. In general, there is nothing in SPIRITS that could not be made with CPL and Parlay, but the main goal of SPIRITS is not to add services that could not be made with other technologies. SPIRITS is mainly for leveraging the present IN platforms so that some IP services can use the present IN. This is important especially in countries where IN is new and the investments are not yet paid off. Therefore there is a strong interest in using the IN platforms as long as they have been planned to be in operation.

The motivation for defining SPIRITS was to provide the following services:

- Internet Call Waiting,
- Internet Caller-ID Delivery,
- Internet Call Forwarding.

3.13 CPL (Call Processing Language)

CPL is one of the XML-based languages created for value added services. Using CPL it is possible to implement simple telephony services, like forwarding and redirecting calls. Voice XML (VXML) is another similar scripting language, for voice controlled services. CPL is defined by IETF and used in SIP-based IP telephony but it is not tied to any particular protocol. In general, SIP uses scripting languages for services in which it

differs from e.g. H.323 which has supplementary services built into the protocol (H.450).

Any XML based scripting language is defined by a Document Type Definition (DTD), which defines elements and attributes. The elements then correspond to states in the script or to methods of some API. CPL like any XML language has a tree structure. It contains on the top level tags *ancillary*, *subaction*, *outgoing* and *incoming*. The actual script is under the last three tags. CPL script contains the following: *switches*, *location modifiers*, *signaling operations*, and *non-signaling operations*. A typical CPL script could for instance use *address* switch to make a decision based on the address, it could retrieve the target location using the location modifier *lookout*, then it could use the signaling operation *proxy* to forward the request to the given location. Additionally the script should handle all error cases, like if the receiver is busy, there is no answer and so on.

XML suits rather well to service creation. However, XML is basically a browser language. It has powerful additions for browser usage, like XSLT, XPath and Xlink. An XML document carries starting and ending tags as ASCII names, also data is in ASCII. Any compression should be on lower layers. For high bandwidth connections this is fine and it helps a browser to display unknown data. However, in XML based data transported over the air interface there is less sense in carrying all tags as full names: both sides of communication must in any case know the methods of the API to be called. In the favor of XML one must admit that features of XML as a browser language are useful in graphical representation and editing of XML scripts.

SIP-CGI is a more powerful scripting language for IP telephony. SIP-CGI scripts can execute arbitrary programs in a SIP server. CGI causes some problems because it is so powerful: poorly used HTTP-CGI has caused serious security problems, like the Windows'2000 IIS bugs.

J2EE Servlets from suitable Java beans package are another possibility, and in this case Java security mechanisms apply. In IP telephony there are proposals for SIP Servlets. In expression power SIP Servlets are similar to SIP-CGI.

3.14 JAIN (Java APIs for Integrated Networks)

The objective of JAIN Community initiative was to create a number of integrated network APIs [10] that abstract the details of networks and protocols implementations and provide service portability, convergence, and secure access to the networks. JAIN integrates service creation in wireless, packet based and wireline networks by separating service-based logic from network-based logic what can be considered as separation into application and protocol layers respectively.

The protocol layer of JAIN is based on Java standardization of specific protocols like SIP, H.323, MAP, etc. It provides the JAIN protocol API that comprises Java APIs for the protocols. As a result applications and protocols stacks can be dynamically interchanged, what can help to avoid current situation when introduction of a new variant of protocol requires upgrading of application-level software.

The application layer provides a single call model JTAPI (Java Telephony API) across all protocols supported in the protocol layer. It is accessible through the Application API that consists of the JAIN Call Control (JCC) API, JAIN Coordination and Transaction (JCAT), JAIN Service Logic Execution Environment (JSLEE) API and JAIN Service Provider APIs (SPA). The two first types of APIs provide interfaces to generic call model supporting access to advanced intelligent network. JSLEE API provides access to a consistent run-time environment and a common model for deploying services in Java, while SPAs provide a Java version of Parlay API.

Service development in JAIN depends on the level of trust the operator of the JAIN environment has on the service provider. Trusted service providers create services using Java servlets and XML, typically Java servlets are given by some Java beans packet. Untrusted service providers access the JAIN environment through Parlay APIs.

JAIN does not build on CORBA, as was the case with Parlay, but on a Java platform: telephony is realized with JTAPI, and different networks can be used under the Java platform. The XML-based technologies, enhanced with Java servlets, can be used for service provisioning for trusted service providers, while JAIN-Parlay APIs are used for untrusted 3pty service providers.

3.15 Some other ideas

TINA: TINA [7] is already an old effort to create a service platform on CORBA realizing the ODP (Open Distributed Processing) model. Though TINA ideas are still continued in the OMG, the architecture's main contribution to future service creation may well be on the level of reuse of ideas. TINA's auxiliary projects, such as Dolmen made successful demonstrations of certain enhancements of CORBA mobility and they may be reflected in CORBA.

Mobile agents: The use of mobile agents (software programs capable of moving in a computer network and acting autonomously on behalf of some entity, however under the control of some authority) can be regarded as a powerful asset for implementing complex and highly dependable distributed software systems. Agents, in fact, may help to better exploit the TINA type

paradigm. This is because the main characteristic of agents, that is to play in autonomy in visited environments with reference to policies set by the originating environment although in respect of constraints imposed by the host nodes, seem naturally suited to model the interactions between components belonging to different business domains. Otherwise stated, mobile agents, in particular, can take along the service logic as well as the user profiles when migrating between different network nodes. It is expected that service availability and flexibility will be considerably increased, while signaling traffic load may be kept to controllable levels. Of course, the price to pay is the availability of an agent execution environment, an “agency”, in all network and user nodes. Additionally, problems of scalability, reliability, security and quality of service have to be faced.

Active networks: Active networks are of two types, active packets and mobile agents. The latter are already explained. Active packets are small programs programming routers or other network nodes. Security concerns of such a proposal are considerable. Active packets are not intended for the type of value added services discussed here. They have some potential, for instance in protection against Distributed Denial-of-Service attacks. Ability to program packet handling mechanisms is useful for services, however, the overhead required to make such a service platform secure may imply packets that are too long.

4. DISCUSSION

There are many technologies for creation of future services and it is easy to be lost in details. Cost, maturity, supported functionality and other criteria are often used for comparison of technologies. This kind of comparison suits poorly to future technologies, which are on an early development stage. We compare the alternatives briefly using three methods: finding the good ideas of the technology, deducing what limits and requirements the set of technologies has, and classifying the technologies into groups.

4.1 Finding Good Ideas

In this method the important part is the good ideas of the considered technology. There are not that many good ideas behind the service creation technologies. Detecting the promising ideas and reasons why the technologies implementing these ideas have succeeded or failed is one way of predicting the future development. That is, good ideas, which are ignored,

rise again in another disguise, and bad ideas tend to end up with similar problems every time they are tried.

Many good ideas have originated in the IN, such as:

- Separation of service logic from basic connection control, and routing to separate service points simplifies introduction of new services.
- External service providers can create services. Although IN did not enable external service providers to offer services on their own platforms but on the operator's platforms, still the idea that there are more roles than operator-customer derives from the IN.
- Services are easier to develop in scripting languages (SIBs in the IN) instead of programming languages as in protocol development. Notice that in TINA this separation was not made. TINA is not used. We see also a similar tendency towards separation of service creation and protocol development in creation of Parlay/X.
- Service creation and provisioning must be a fast process in a competitive situation.
- Capability sets and targeted services are a way to develop service creation platform in phases.
- Personal mobility in the form of UPT is a promising idea.

CAMEL has basically two good ideas:

- Not to reinvent a wheel, IN is a good service creation technology.
- Successful IN services can be made by alternative charging (prepaid).

Some ideas derive from WWW:

- Use of markup languages for service creation (HTML, WML, XML).
- Use of scripts (CGI, CPL), Java applets, Java servlets to enhance the services.

Very importantly, the Internet and WWW showed that successful end-system data services are possible, but here we concentrate on value added services using network resources.

A few ideas were originated in ODP and TINA:

- Use of CORBA platforms, continued by Parlay.
- Dynamic service discovery.
- Importance of the business models as an elaboration of the IN three roles (operator, subscriber, and user). The business model comes from the enterprise viewpoint of the ODP design process and is included in UML

as use cases. However, the business aspect of the business model is a very important and often understudied concept for new services.

Parlay and OSA have elaborated the idea of an external service provider, already clearly defined in TINA:

- External service providers can be allowed to access network resources via secure APIs.

WAP, MExE, J2ME have one good idea:

- Mobile networks are a success, the Internet is a success, let us merge them and it should be a success.

VHE has elaborated the idea of Personal Mobility to:

- User should be able to use all services as if he was in his home environment.
- Single-sign-on can be understood as an idea related to the Personal Mobility and VHE.

The idea in Mobile agents is autonomous agents forming an agent community where the members are discussing with each other in agent languages, This AI idea has not been realized and the technology is not in the among the most probable service creation methods, mostly due to security considerations. Active packets of active networks have even more serious security concerns than mobile agents.

There are also some doubtful claims about services in which many strongly believe:

- There are many useful services to be invented that users will pay for. There actually are some successful services: IN FreePhone, PremiumRate, VPN, Televoting, CAMEL Prepaid, Internet WWW, email, file transfer, but most services find few users.
- There are some ideas that sound good: brokering of services as a business idea, eCommerce, mCommerce, VHE. The doubtful claim is that they are good enough.

4.2 Limitations And Requirements

End system data services, like the Web, are proven. Many of the service creation technologies try to realize commercially successful value added services, which use network resources. This idea used to be alien to the

Internet. There are limitations arising from this goal. Value added services using network resources are mostly build on the following possibilities. Changing routing (examples: IN VPN), modifying charging (examples: IN FP, PR), avoiding assumptions in network dimensioning (example: televoting), providing anonymity to the receiver or to the caller (examples: IN PR), adding specialized resources (examples: answering machines, DTMF), locating users (examples: Camel locate in Parlay), finding information (examples: WWW search machines, brokering). These enable a large range of services, but not necessary to the extent that is often thought. Most new good value added service ideas are economically failures when they are tried.

As for requirements, there are some coming from the goals of service creation. For competitive reasons services should be created by external service providers in a very short time. This means that the service creator cannot be expected to write a secure service. It must be the service platform that takes care of security, management, charging etc. supporting activities. This is a positive shift from the Internet style of protocol development directly on top of the socket interface on the weakly typed C-language using Unix system calls running as root, having given us all root exploit problems in the past. If the service platform is secure enough, maybe the bug problem can be solved. However, the mechanisms to make a secure platform, like sandboxes and access rights given to authenticated code, are not sufficient as they restrict the services that are possible. Special attention should be paid to standardizing the requirements of a service platform and to specification of new tools which can check that the requirements are filled. Security and reliability are one part of the requirements.

4.3 Classification

Main stream future service creation technologies seem to fall into three groups: IN-based services, Parlay-based services and MExE-based services. These groups are likely to coexist in the future hybrid network and they can be seen as complementary.

The IN-based technologies include IN, CAMEL and PINT/SPIRITS. IN is a mature service creation technology and CAMEL will very likely be heavily used in 3G. In this scenario, the IN is in the PSTN side and seems like a technology that will disappear, but one should not forget IN CS4. IN is actually developing towards a service creation technology for hybrid networks.

The services in the operator's side in all-IP networks are likely to use Parlay APIs. The technologies Parlay, OSA and TSAS all might converge to

Parlay. VHE is a future goal of these technologies and it may mean a great change in the service field. For the time being, VHE is still not mature enough. JAIN can also be included in this group, though JAIN is considered as a technology for hybrid networks. JAIN and Parlay/X combine XML-based service creation to this set of technologies. We should expect to see Parlay API, Java servlets and XML-based service languages in the IP network.

The last set of service creation technologies concentrates on services to mobile equipment. WAP and J2ME may be deployed alone, but in 3GPP they are combined under the MExE technology. MExE actually adds only some security features to these technologies, but it is likely that MExE will be used. There is no competition between the MExE set of technologies and the Parlay set of technologies. Because of CAMEL and PINT/SPIRITS there is also no conflict with MExE and IN-based technologies. There may be some competition with Parlay type ideas and IN/PINT/SPIRITS, though the first suits to All-IP better and the latter is essentially a solution to hybrid networks.

These three sets of service technologies can bring a mobile, IP or PSTN user value added services which resemble the services offered by IN and WWW. The role of the remaining technologies is to go further: IN and WWW are still very limited in their services. Ideas, such as brokering, trading, auction, intruder response, management and so on will require new technologies. This is where mobile agents, active networks, old TINA ideas and other technologies that seem to be off the main stream come to play a role.

This paper has presented very briefly the main service creation technologies and illustrated some approaches for their comparison in an early stage. Service creation is only one aspect of service development, there are also service provisioning and service management solutions. Due to the limited space and time we cannot discuss these technologies, but they should be considered when comparing the service technology alternatives.

Acknowledgement

The authors want to thank their colleagues and students for fruitful discussions that made the paper possible.

5. REFERENCES

- [1] I. Faynberg et al, "The Intelligent Network Standards: Their Application to Services", McGraw-Hill, 1997
- [2] S. Petrack, L. Conroy, "The PINT Service Protocol: Extension to SIP and SDP for IP Access to Telephone Call Services, Proposed Standard", RFC 2848, June 2000
- [3] L. Slutsman, et al, "The Spirits Architecture", draft-ietf-spirits-architecture-03.txt, February 2001
- [4] ITU-T, Recommendation Q.1241, "Introduction to Intelligent Network Capability Set-4", 07/01
- [5] MExE Forum website <http://www.mexeforum.org>
- [6] Parlay Group WWW-home page <http://www.parlay.org>
- [7] TINA-C website <http://www.tinac.com>
- [8] WAP-169-WTA-20000707-a. Wireless Telephony Application Specification.
- [9] WAP-190-WAESpec-20000329-a. Wireless Application Environment Specification.
- [10] website <http://java.sun.com/products/jain/>
- [11] 3G TS 22. 038 "USIM/SIM Application Toolkit (USAT/SAT); Service description"
- [12] 3G TS 22.121 "The Virtual Home Environment"
- [13] 3GPP TS 23.078, "Customized Applications for Mobile network Enhanced Logic (CAMEL)", March 2001
- [14] 3GPP TS 23.127 "Virtual Home Environment/Open Service Architecture", June 2001.