NetApp: Autonomic-network-based application architecture for creating new value-added services

Tuan Loc NGUYEN[1], Francine KRIEF[2], Abbas JAMALIPOUR[3], Guy PUJOLLE[1]

[1]Laboratory Lip6, University Paris 6
8 rue du Capitaine Scott, 75015 Paris, France
Email: {tuan-loc.nguyen, guy.pujolle}@lip6.fr
http://www.lip6.fr
Phone: +33 1 44 27 87 85

[2]LaBri, University Bordeaux 1, France
Email: francine.krief@labri.fr

[3]School of Electrical and Information Engineering
University of Sydney, Australia
Email: a.jamalipour@ieee.org

Abstract - The generalization of Internet continues to grow unabated, especially with the rapid adoption of the mobile and fixed networks connecting a myriad of devices at home and on business. This allows acquiring more and more services at low costs. In the future, anyone with any kind of devices at any times will be able to access any type of service from anywhere. To fulfill this requirement, new rapid and profitable methods for service creation must be investigated.

In traditional telecommunication networks, the intelligence of a service has been for so long achieved at the heart of the network. The Internet intermediation involves moving the intelligence towards the edge of the network. A new and fresh look is needed to design and develop self-organizing networks so that human intervention is minimized as much as possible.

In this regard, we propose the autonomic-network-based application concept, NetApp. This is an architecture in which the applications and the services are not supported by a pre-existing network, but where the network itself grows out of the applications and the services that end users require. The major objective of this article is to propose a new architecture for the adaptive and self-organizing ambient-aware applications and services in networking.

Keywords: Value-added service, autonomic-network-based application, intermediation, XML, IP telephony.

Paper Topic Areas:

- Adaptive and self-organizing ambient aware applications, services and middleware,
- Self-organization and reconfiguration of the transport network to meet the individual and collective needs of applications and services,
- Automatic adaptation of protocols and software to meet the desired goals and objectives.

NetApp: Autonomic-network-based application architecture for creating new value-added services

Tuan Loc Nguyen, LIP6,

Francine Krief, LABRI,

Abbas Jamalipour, Sydney Univ.,

Guy Pujolle, LIP6.

Abstract

The generalization of Internet continues to grow unabated, especially with the rapid adoption of the mobile and fixed networks connecting a myriad of devices at home and on business. This allows acquiring more and more services at low costs. In the future, anyone with any kind of devices at any times will be able to access any type of service from anywhere. To fulfill this requirement, new rapid and profitable methods for service creation must be investigated.

In traditional telecommunication networks, the intelligence of a service has been for so long achieved at the heart of the network. The Internet intermediation involves moving the intelligence towards the edge of the network. A new and fresh look is needed to design and develop self-organizing networks so that human intervention is minimized as much as possible.

In this regard, we propose the autonomic-network-based application concept, NetApp. This is an architecture in which the applications and the services are not supported by a pre-existing network, but where the network itself grows out of the applications and the services that end users require. The major objective of this article is to propose a new architecture for the adaptive and self-organizing ambient-aware applications and services in networking.

I. INTRODUCTION

The development of Internet paves the way for the introduction of more and more services such as e-newspapers, e-shopping, etc. People will be able to access any type of service from everywhere via any kind of devices [1] (PDA, mobile phone…). This perspective will create not only better but also low cost services for consumers.

At the beginning stage of the Internet era, people believed that the development of Internet will eliminate the need for an intermediary. In other words, producers and customers will be able to interact directly through the Internet.

Progressively, the amount of information transported through the Internet grew rapidly. This yielded to the fact that people are unable to identify which information is important, trusted and how to easily find it. So in the model of net economy, the intermediary Internet reappears at a higher level compared to that of the past. Furthermore, it has new roles such as the provision of product information to buyers and marketing information to sellers, the aggregation and the management of both physical deliveries and payments, providing trust-based relationships and ensuring the integrity of the markets. As a result, the consumer and the producer not only interact directly but also through mediation services. This creates value-added services and can aggregate service information directly between producers; generating thus more value [2]. Fig. 1 illustrates the intermediation concept in service creation.

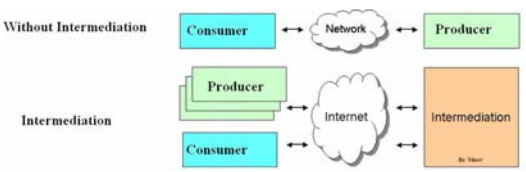

Figure 1 - Intermediation concept

To achieve the aforementioned perspective, autonomic-network-based application service architecture (NetApp) is proposed. Its purpose is to easily create a plethora of better services. In the NetApp architecture, many protocols such as LDAP databases, IP telephony and Web services are integrated into one protocol in the same IP infrastructure, in order to reach its purpose without difficulty. The concept is that we can separate the service logic from the others and we only need to create new services but don't care about how the network structure is. The intelligence of service, once achieved at the heart of the traditional telecommunication networks, is being moved, thanks to the advancement of the Internet intermediation, towards the edge of the network. Fig. 2 illustrates the evolution of the Intermediation in service creation.

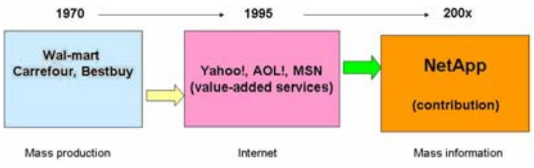

Figure 2 - Evolution of Intermediation concept

The organization of the article is as follows. In session II, we present an overview of the state of the art about service creation in networking. Then in session III, we present the NetApp architecture for value-added service creation. The analysis of the NetApp architecture is discussed. We provide in session IV, a testbed and a corresponding evaluation of this testbed. Finally, session V concludes the paper with some perspectives related to the NetApp architecture and the future work to be done.

II. Related work

Developing information technology over the last ten years has moved rapidly from single use, centralized systems to distributed, multi-purpose systems which are now increasingly embedded in a fully interconnected network.

We can see through machine readable content (Web semantic), novel system-system interaction models (peer-to-peer computing), dynamic discovery and description of web enabled software applications (Web services), XML-based business process infrastructures (e.g., Sun ebXML and Microsoft SharePoint Portal), future development environments based on XML standard (Sun ONE and Microsoft .NET). Technology development by itself however is not enough to bring about the vision outlined in the introduction. In particular, this new generation of on-line systems must deal with the challenges raised by the deployment in a truly open environment like our proposal architecture, NetApp. In particular from our experience, we believe there are two categories that can be close to provide the *open environment of service creation in networking*:

1. Autonomic and interoperable communication:

That is, how to enable on-line software systems to interact with one another in increasingly flexible ways: configurable interaction sequences, determine the intended meaning of any message or datum in the context of its usage. This requires the coherent integration of technologies from traditional knowledge representation, databases, semantic models (Web semantic), connector communication languages and protocol formalisms (LDAP, Simple Object Access Protocol SOAP, etc). Fig. 3 illustrates the evolution of n-tiers architecture to satisfy the autonomic communication:

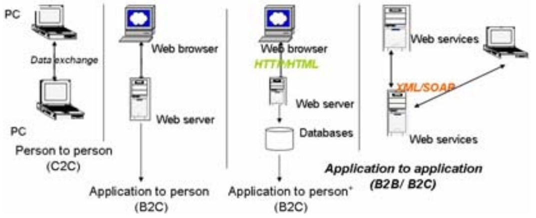

Figure 3 - Evolution of n-tiers architecture

a. OSGi:

The Open Services Gateway Initiative OSGi [3] aims at specifying a service gateway in order that all types of terminal that can coordinate and cooperate. Its target environments are mainly home networks. It can be a gateway towards the Internet and has the needed APIs to provide a tool for deploying new services in home networks. The OSGi provides a platform for service delivery and has been designed to host services so that the services can be located from the provider. But in the future, there will be a lot of service providers, a single home may be served by numerous providers. And a new service maybe a combination of different services from several providers and it will be a whole unit, i.e. very hard to be split.

b. Parlay and 3GPP:

The Parlay group defines a set of abstract APIs in a programming language neutral way. Parlay ensures that a developer who is not necessarily telecommunication specialist can develop new and innovative services. The framework 3GPP provides applications with the basic mechanisms that enable them to use the services offered by the networks. First, an application can use the network functionality that is available by Parlay API. After authentication, the discovery function enables the application to find out which network service provided by Parlay APIs.

However, mobile networks based on 3GPP are very powerful but inflexible. This means that roaming a service between two networks is also established manually, resulting in good internetworking but only for pre-defined, fixed-services such as SMS or basic data services. Parlay and 3GPP also need to be more flexible in selecting of the programmable capabilities. Parlay APIs are too rigid in definition to allow the service provider to offer a new service.

2. Coordination and automation for service creation in networking:

That is, understanding how to effectively automate systems in an open environment and putting in place frameworks that enable automatic creation, maintenance, execution and monitoring of contracts and agreements between automated systems to fulfill their business objectives, even with unknown parties and without the necessary supervision of human operators. This would enable the dynamic creation of supply chains, virtual companies and other organizational structures for the fulfillment of user and business goals. Successful systems are likely to require the integration of security, management, multi-agent organization techniques such as negotiation and distributed problems solving with existing or future industrial strength e-business suites.

a. Intelligent protocols for service creation in networking:

Besides the native network functions, such as routing and packet forwarding, future service architectures are required enabling location and utilization of services. A service is a program, which can be accessed about standardized functions over a network. Services are allowed cascading without previous knowledge of each other and thus this enables the solution of complex tasks. A typical service used during the initialization of a client is the localization of a data. Gateways or neighboring nodes can provide this service. To find this service, the client can use a service discovery protocol. Many protocols already provide facilities to make networking easier for end-users thanks to service discovery (UPnP, Bluetooth, SLP and Zeroconf). They are usually neither implemented nor effective (e.g., Java Virtual Machine needed). Above all, there are vendor-specific solutions that prevent them to work together. SLP and Zeroconf are, at present, the most used protocols. Both protocols are quiet similar about the way service discovery is running. Unfortunately, neither Zeroconf nor SLP proposes a mechanism for adapting services to the context.

Flexible application-layer for autonomous device interaction which is supported on personal computers is at its beginning stages. Universal plug-and-play (UPnP) [4] is one of the most established middleware platforms for this purpose. UPnP defines a set of universal, open, XML-based protocols that allow the semantic description and control of various services. Unfortunately, the technical realization of the UPnP is complex. The client that supports the entire UPnP protocol is thus power demanding and expensive, so that many small and thin clients will be excluded from the integration into such networks.

b. Web services and service composition:

A Web service [5] is a composant based on XML standard. The interface is described by WSDL language (Web Service Definition Language). The transport is provided by SOAP protocol (Simple Object Access Protocol) and can easily be found by a yellow page named UDDI (Universal Description, Discovery and Integration).

Different elementary Web services can be composed [6] to provide a new Web service, which combines their functionalities together. The classical example is the travel agency where a new service is proposed by composition of Web services (hotel, train, plane reservation). The workflow "Web-Service-Business Process Execution Language" (WS-BPEL) is the composition language proposed by the W3C. Using this language, different compositions are possible such as calling to Web services:

- In sequence or in parallel;
- With or without synchronization constraints.

Web services have recently emerged as a promising technology for providing the standard interfaces between the software composants by using XML to encode a variety of information for the Web. Client and server of a Web service can be invoked and run on different platforms. On the other hand, Web services don't have enough interoperability that will arise with integrated networks like IP networks and telecommunication networks. Moreover, Web services miss a real link language, a universal assembly language, to create new services by combining numerous Web services together.

III. NETAPP: AUTONOMIC-NETWORK-BASED APPLICATION ARCHITECTURE

There are three main motivations for using NetApp architecture. First, our work is in line with autonomic communication, in which the applications and the services are not supported by a pre-existing network, but where the network itself grows out of the applications and the services that end users require. Second, NetApp aims to create an open environment based on existing standard technologies that not only supports multiple client types, but also adapts to almost all types of services, from the IP telephone services to the Internet services. Third, this work may potentially become a junction between two research disciplines in computer science: software and networking.

1. NetApp architecture

NetApp, autonomic-network-based application architecture, aims to propose a new method for creating rapidly new services. NetApp springs from merging the intelligence in networking concept into the intermediation concept. In the intelligence in networking concept, the service logic is independent of the basic function (the switching, message transferring, packet routing, etc). The Intermediation concept (e.g., Yahoo!) preconizes the federation of back-end services. This is also the access point to multiple services over heterogeneous networks and from any type of terminals. NetApp is illustrated in Fig. 4.

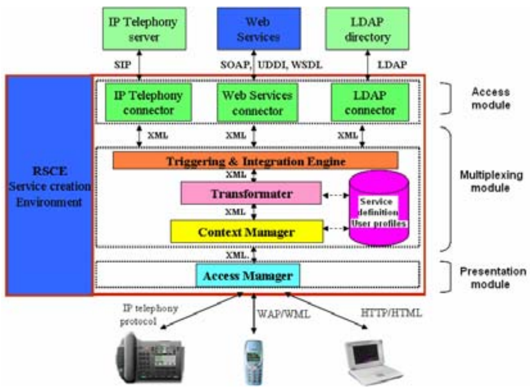

Figure 4 - NetApp architecture

NetApp architecture is the three-tier architecture including: the access module, the multiplexing module and the presentation module:

a. Access module

The access module comprises the connectors [7] respectively associated to the dedicated servers with which they are able to dialogue. Each connector includes the rules of treatment depended on the service provider. For example, the connector (IP telephony, LDAP or Web services) interacts with external heterogeneous applications and converts the data of these applications into an intermediate language, XML. The SIP protocol can be used as a basic platform for service creation in many regards in the IP telephony server. The goal is to create a unique format at the intermediation level to unify the heterogeneous services.

b. Multiplexing module

This is a virtual execution machine composed of the triggering & integration engine, the transformation and the context manager. This generates value-added services from the basic services. Thanks to the intermediate language XML, the modules in the portal manager can easily communicate together.

The triggering & integration engine is triggered by events. For each page received from the connector module, the engine invokes the connectors according to the matching service definition. Furthermore, new value-added services can be defined as a sequence of services. This will be processed in a pipeline managed by the integration engine. The triggering realizes the interaction between the different services managed by the dedicated server. It can trigger the requests coming from the presentation module to the access module and transport the information from the access module to the presentation module. It manages in addition the open sessions with the different terminals managed by the presentation module.

The transformation enriches and transforms services into a value-added service. The transformation is based on services description that is managed by the *service creator* [8]. The transformation makes the interface with the dedicated server corresponding to the considered access module. It manages the whole dialogue with the dedicated server and transforms the received information from the server into a common format that corresponds to the exchanged page of the server. To do that job, it has a translation table created by the service creator. This translation table enables the transformation to associate the exchanged information according to the specific format with the dedicated server. Then, the received information is transmitted to the context manager to create a dynamic page to the client or to another server of the access module to trigger a new service.

The context manager gives the user single personalized access to relevant information and services, as well as single point of interaction with these information and applications from multiple devices. The goal of the context manager is to create personalized logical pages (i.e. logical screens or logical windows or folders) that provide individual user relevant personalized information. For example, it personalizes the user's view with relevant information and manages user interactions with integrated applications according to the user profile: rights, priorities and permissions.

c. RSCE (Rapid Service Creation Environment)

In NetApp architecture, there is another important module, the RSCE [9], which aims to create rapidly new services and deploy it to the network. The NetApp components like the triggering and integration engine, the transformation and the context manager are built by assembling the composants in the RSCE, an environment for service creation. The RSCE deploys the new service through the service definition module. The integration engine can assembly these services described by a script written by the service creator in RSCE and executed on the integration engine (Fig. 5):

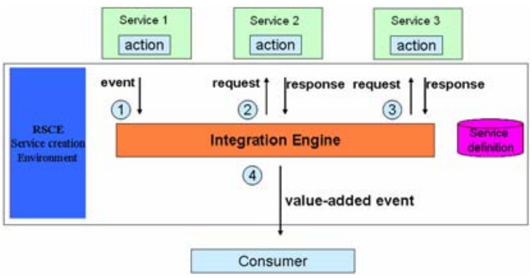

Figure 5 - Integration engine in NetApp

An invocation of classical service or a function call can be described in XML, a meta-language of the Internet. Here is an invocation service in NetApp:

```
<methodCall>
    <methodName>...</methodName>
    <input_Params>
        <param>
            <value>…</value>
        </param>
    </input_Params>
    <QoS_Params>…</QoS_Params>
    <output_Params>…</output_Params>
</methodCall>
```

The XML procedure calls are particularly effective for the automatic sequence of the treatments in which some XML parameters or XML tags at one output procedure can be used by another procedure. The integration engine can also trigger the procedures according to the definition of the NetApp added-value service, described in XML.

```
<NetApp_Service>
    <matched_Event>…</matched_Event>
    <methodCall>…</methodCall>
    <methodCall>…</methodCall>
</NetApp_Service>
```

The value-added service in NetApp can be defined, tested and deployed by the service integrator. The developer does not care about how the integrator did it.

d.Presentation module

This module adopts the thin client computing protocol and can format the virtual display depending on the terminal type. It receives each page from the multiplexing module with a subscriber who is using this terminal in order to generate the adapted information to display on the terminal starting from the interaction elements in the received page from the multiplexing module.

The presentation module can also interact with several types of devices. The terminal becomes the thin device; it focuses more on displaying messages sent by the presentation module. The presentation module sends to terminal the informative (user interface) messages, which should be followed by appropriate action invocations. These messages allow a set of services to be displayed among which users would like to choose.

2. NetApp analysis

Communications is a very large subject. Computer communications can transfer data networking while telecommunications can transport the public voice networking. However, both domains are now rapidly converging into the same network to support all types of services such as data, voice, Web, etc.

The Internet is opening the classical model adopted by telecommunication companies. The telecommunication networks are migrating toward Internet technology. Former protocol barriers between networks, terminals and servers are subsequently removed. With the advent of the Internet, telecommunications and information technology services share the same IP infrastructure. Services of the future are no longer only telecommunications or data services. Services of the future will be unified services combining telecommunications and data services. The NetApp architecture has the potential for aggregating telecommunications services and data services. Thus, this architecture can generate more value and create new opportunities and new markets.

For example, with the classical PABX call server, when receiving an incoming call, the callee will only receive the caller’s phone number, that’s all. Now, with the IP telephony server, an incoming call arrived, via the SIP-IP telephony protocol, the integration engine in NetApp can interrupt the treatment process and triggers to the Web services for finding the name or address corresponding to this call. Then, the callee will not only receive the call number but also more detailed information about the caller.

IP telephony should be cheaper, better or different from a traditional telephony. IP telephony can be regarded like a service running on top of IP infrastructure. It includes the voice over IP and data over IP. The voice over IP is independent of network-level signaling. It uses the IP infrastructure for transporting the signaling messages and the encoded voice. The data over IP is a solution to combine IP telephone services with other traditional services of the IP world, such as Web services or electronic messaging, thereby sharing the same IP infrastructure.

Furthermore, NetApp environment provides full mobility capabilities, terminal mobility, user mobility and service mobility: allowing the user to telephone or access various services such as surf on the Web, receive email notifications, alarm notifications in addition to warning systems connected to the owner’s home or car. Therefore, in NetApp architecture, the major developments in communications technology have been driven by the services that can be offered to the end-users; they don’t care about technology, value is in the service.

IV. Testbed and evaluation

1. NetApp testbed

In this session we present a global platform that we are deploying to test a new value-added service from NetApp architecture. This platform, illustrated in Fig. 6, is composed of IP telephone terminals, a PC with HTML browser, an IP-PABX call server, a Web server and a NetApp server.

One of the goals of this testbed is to validate and experiment the NetApp architecture to create rapidly new value-added services in networking via RSCE environment (in NetApp server). In this testbed, we realized three phases: a phase of integration from the different components of the testbed, a phase of service creation by the assembly of basic functions of the NetApp architecture and finally a phase of NetApp analysis. In addition, we also expect to get more performance evaluations such as the measurement of response times, deployment times on the facility as well as the speed of creation of added value services.

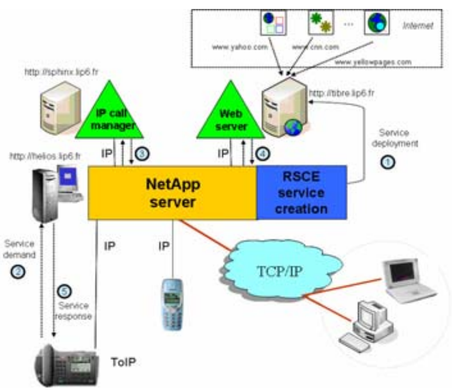

Figure 6 - NetApp testbed

In the phase of service creation, to assembly the functions in NetApp server, each connector in NetApp server has to publish its available services description and a dictionary (rule of service transformation) in a XML file named list_services.xml:

<services timeout=1>
<service name="yahoo" link="www.yahoo.com"/>
<service name="news" link="www.cnn.com"/>
<service name="yp" link="www.yellowpages.com>
<dictionary link="http://helios.lip6.fr/tranformation_rule.xml"/>
</services>

In every predefined interval time, by reading the configuration file (externalservers.xml), the NetApp server (http://helios.lip6.fr) can check the list_services.xml file from every external server (via their corresponding connectors):

<servers>
<server name="ip_pabx" link="http://sphinx.lip6.fr"/>
<server name="web" link="http://tibre.lip6.fr"/>
</servers>

In order to limit coding development in RSCE, the new services can be provided by another provider, that means it can be developed apart from NetApp server and can be accessible transparent for the client. In this case, the RSCE can be implicitly implemented in the NetApp server. The services can be stored by a Web server in the network or in a local/network directory that can be accessible from the NetApp server. The NetApp server is able to reach these servers through the interface provided by the service creator in every external server. These services are developed by using XML language and can be static or dynamic (pages generated by PHP scripts in this case). The user such as IP telephone terminal can request any external service by the “services” key already used to access to the basic services of the NetApp server.

2. NetApp evaluation and benefit

Through the NetApp testbed implementation in Fig. 6, we believe that our proposal can be a critical factor in the development of next generation open service environments. Many technologies in NetApp concept were tested in a truly open environment and will be migrated from this research environment into exploitable commercial propositions. We also believe that a crucial part in the development of the next generation environment can be played by combining pure and applied research with business usages. Furthermore, the combination of heterogeneous technologies from these spaces to create viable environments, which support high levels of autonomy, flexible system communication and advanced dynamic composition of services, remains a great challenge.

The expected benefit of this testbed includes the following:

- Automated highly *proactive* and intelligent services have the potential to reduce the operational cost of systems and improve access to any service.
- Increasingly flexible *interoperability* between online services has the potential to promote integration within supply chains, reduce the operational cost of systems and promote accessibility from various devices and contexts.
- The techniques for *service creation in open environments* could be improved to reduce the time required to develop systems.
- Wide-scale deployment of advanced network *components* and services across a large number on an advanced prototype basis like IP telephony, LDAP, Web services can be tested in NetApp concept.

Furthermore, through NetApp architecture, we expect that telecommunications operators or service providers can also differentiate themselves in a competitive market by their value-added in the services offered. Thanks to the NetApp, the number of value-added services resulting from this combination can grow in an exponential way [10].

V. CONCLUSION

NetApp is an open research program focused on a common project vision and agreed strategic goals. NetApp would build on existing standard technologies and aim to make very significant contributions to the evolution of these through wide-scale deployment experience in innovative applications in future networks. NetApp facilitates the programming on high abstraction layers. The presentation of NetApp, leads us to the facility of rapid creation of value-added services. This architecture adapts to almost all of the requirements, from the IP telephone services to the Internet services.

Based on the requirements an autonomic-network-based architecture, this article explores a junction on the frontiers of two research disciplines in computer science: software and networking. Software has long been concerned with the categorization of software designs and the development of design methodologies, but has been rarely able to objectively evaluate system behavior. Networking is focused on in the communication between systems and improving the performance of communication techniques. NetApp is the junction between software and networking.

Moreover, it is very important to maintain the QoS offered to the users during the terminal mobility. A management architecture can be proposed to offer a personalized access to relevant information and services or applications, which guaranties a good quality of service. The software components can be chosen by taken into account the users' QoS needs. With this approach, contractual services (via a service level agreement) can be allowed. The management architecture proposed can ensure the maximum automation of the activities, the maximum autonomy of the components entering in the services offer. This approach, including a policy management environment and an expression of services contracts, will guarantee a complete automation of the services offered in networks.

REFERENCES

[1] A. Fasbender and al., "Any network, any terminal, anywhere," *IEEE Personal Communications*, vol. 6, pp. 22 – 30, April 1999.

[2] R. Barrett, P.P. Maglio, "Intermediaires: An approach to manipulating information streams," *IBM Systems Journal*, vol. 38, no. 4, 1999.

[3] OSGI, "OSGI service gateway specification," release 2, October 2001.

[4] Microsoft Corporation, "Universal Plug and Play Device Architecture 1.0," *Microsoft Corporation*, USA, 2000.

[5] D. Booth and al., "Web service architecture", published document of Web services architecture working group W3c, February 2004.

[6] C. Szyperski. "Component technology - What, where, how," *ICSE 2003*, USA, pp.684–693, May 2003.

[7] R.T. Fielding, R.N. Taylor, "Principled Design of the Modern Web Architecture," *ICSE 2000*, Ireland, pp. 407-416, 2000.

[8] T.L. Nguyen and al., "Network-based application architecture," *RIVF2005*, Vietnam, 2005.

[9] T.L. Nguyen, T.A. Nguyen, "SCDL for IIS architecture," *Cosci2005*, 2005, Vietnam.

[10] T.L. Nguyen, "Architecture of Intelligent Semantic Services (Archises)," *PhD dissertation*, 2004, University Paris 6, France.

BIOGRAPHIES

TUAN LOC NGUYEN (tuan-loc.nguyen@lip6.fr) received his Ph.D. in network and computer science from the University of Paris VI, France, in 2004 and his engineer degree in science computer from Polytechnic University of Ho Chi Minh City, Vietnam, in 1999. He is currently an assistant professor at the University of Paris XII and Paris VI. His research interests include the intelligence in networking, value-added services in networking, service provision, network-based application and autonomic communication.

FRANCINE KRIEF (francine.krief@labri.fr) is currently a professor at the University of Bordeaux 1, France. She is responsible for a research group named CoMet (Context-aware Management and Networking) at the LaBRI laboratory.

ABBAS JAMALIPOUR (a.jamalipour@ieee.org) has been with the School of Electrical and Information Engineering at the University of Sydney, Australia, since 1998, where he is responsible for teaching and research in wireless data communication networks, wireless IP networks, network security and satellite systems. He is a Senior Member of the IEEE, a Fellow Member of IEAust and a Distinguished Lecturer of the IEEE Communications Society. He holds a PhD in Electrical Engineering from Nagoya University, Japan.

GUY PUJOLLE (guy.pujolle@lip6.fr) received his Ph.D. and These d'Etat degrees in computer science from the University of Paris IX and Paris XI in 1975 and 1978, respectively. His is currently a professor at University of Paris VI and a member of the scientific council of France Telecom. He is chair of IFIP Working Group 6.2 on Network and Internetwork Architectures. He has published widely in the areas of computer systems modeling and performance, queuing theory and high-speed networks. His research interests include the analysis and modeling of data communication systems, protocols, high-performance networking, intelligence in networking and wireless networks. He is a Professor Honoris Causa of Beijing University of Posts and Telecommunications since 1988.