Service Definition, Deployment, Management and Architectures in Telecommunication Networks: From Intelligent Networks to Service Oriented Architectures

Sarandis Mitropoulos
Visiting Lecturer
Department of Informatics
University of Piraeus
Piraeus, Greece

Christos Douligeris
Associate Professor
Department of Informatics
University of Piraeus
Piraeus, Greece

Abstract

The rapid globalization of world enterprises and the pressing need to provide networks and services more responsive to the "end users" push the telecommunications world towards rapid service deployment and new integrated value-added technology solutions. Service orientation, initially introduced in Intelligent Networks and currently further exploited within the Service Oriented Architectures (SOA), seems to be the key for achieving the rapid development of new services with low cost. New value-added services as well as new business models drive the adoption of SOA in the telecommunications world. In this paper, we first discuss the requirements for a service-based approach, and then we overview the traditional service orientation in the telecoms sector, as well a main methodology for service definition and creation. A discussion of telecommunications service management follows together with an investigation of current Service Oriented Architectures, through the provision of some important examples.

Keywords: Service Definition and Creation, Service Oriented Architectures, Intelligent Networks, Service Management

1. Introduction

Telecommunication networks of various types such Wireless (i.e. WLAN, LMDS), Satellite (i.e. VSAT), High-Speed Metropolitan (i.e. FDDI), 3G mobile (i..e. UMTS), self-configured (i.e. Ad-Hoc), etc. provide the users with a wide range of value-added services (VAS), virtual private networks, prepaid card calling, web-based portal applications, high-speed last mile access, etc. At the network level, the add-on service level and the application level the design of telecommunications follows a service-based approach. For this reason, the design of telematic systems has been mainly driven by service orientation as a useful method for handling several issues. In the telecommunications framework, the service creation has become a separate system component for embedding flexibility and intelligence within the network.

Due to the existing strong competition between enterprises, the existing strategic alliances, mergers and acquisitions as well as the demand forced by new VAS services, the industry is pushed to achieve their rapid development with low cost goals. Furthermore, the enterprises cannot easily abandon the existing high cost deployed systems. One solution to this problem, that has been followed by some Service Providers, is to outsource some of their infrastructure to appropriate trusted third party providers, focusing in this way more on their core service provision business. However, outsourcing may require the integration of the systems of the participating entities, something that consequently requires appropriate interconnected gateways, while the service provider may lose some control over its service provision.

These requirements force the development of techniques to remedy the problems that arise. Service Oriented Architectures (SOA) seem to be a very promising approach for the telecommunications and the add-on service providers providing them with flexible configuration of new services for emerging markets. SOA also come as a solution to reusability and migration problems. The traditional service-based approaches have been proved inappropriate to support such tasks. Instead, SOA platforms are able to connect multiple technologies, to consolidate multiple services and to provide integrated management. Service oriented middleware technologies provide flexible means for solving the above problems and supporting the next generation networks with service delivery platforms [1].

The goal of this paper is to provide an integrated presentation of all the main issues that are faced by service-oriented approaches in telecommunications and VAS systems. Thus, first we provide an overview of the traditional service-based approach that was incorporated mainly in the Intelligent Networks. Then, we present service definition methods both at the application and the communication levels, in order to provide a picture of what a service at these levels consists of. We also present service provision and a service management integrated framework based on TMN. We continue by presenting several SOA architecture implementation issues by giving certain SOA based examples such as SOCAM, UMTS OSA and OSGA.

2. Traditional Service-based Approaches in Telecommunications

A major component of telecommunications service provisioning is the service Creation, Deployment and Management environment (CDME) that enables the service creator and provider to define, design, develop, test, manage and maintain the service in a rapid and reliable way.

From the service definition point of view, users interact with the telecommunications system endpoints that interact consequently with the core telecommunication service for the purpose of achieving a task-specific goal. Thus, the characteristics of telecommunications services as well as the interactions between the services and the users influence the usability of the provided services [2].

Traditionally, the service-based approaches in order to be effective and flexible with respect to the service design and creation are usually based on the concept of the decomposition of services into reusable and network independent building blocks, called service-independent building blocks *(SIB)*. Each building block implements a required functionality. The building blocks can be represented as a set of graphical icons creating in this way a graph suitable for making decisions.

The main point for service creation is that the service must provide the user with the global service logic that consists of the required service independent building blocks. The latter are accompanied with the service relevant data. Continuous verifications must take place in order the service creation to be efficient with a minimum number of backtrackings. Of course, a suitable management system must undertake the deployment of the service logic as well as the respective data over the network. The overall system endpoints must provide the subscribers with appropriate access rights in order for them to be possible to configure their service profile and functionality [1].

It becomes clear that this traditional service based approach that has been for at least the last fifteen years needed in telecommunications, complicates the infrastructure for the management systems as well as for the distributed applications running on top of them. In fact, these services form large-scale distributed systems, as seen by the both viewpoints the telecom service provider and the telecom service user. Distributed system infrastructure that includes computers, raid disks, networks, middleware components, etc. must perform efficiently with reliability, availability and security. The latter drive the need for an integrated network, service and application management.

Thus, the main goals of the traditional service-based approach were quick time to market, low-cost deployment and high quality of service. Even though it is ambiguous whether the approach succeeded adequately, it provided the industry with useful experience and heritage for new developments such as the service oriented architectures that offer open service interfaces for enabling interoperability between existing heterogeneous networks, services and systems, and for building new system and services *"Just-In-Time"* [3].

3. Overview in Service Definition

In this section, we give a brief presentation of the characteristics of a telecom service. This is necessary in order to understand how a service can be deployed and managed. In general, a telecommunications system consists of a set of communicating entities that communicate via communication channels. The behavior of such a system is usually described in terms of the means, the qualities and quantities that exist on a communication channel during a communication session, plus the control functionality provided to the communicating entities. Controls must become available to a communicating entity in order to pass messages that can be interpreted by another system. Interpreted messages can change system attributes. Thus, the controls are very useful because they enable the users to change media properties or the state of other communicating entities [4].

From the user perspective point of view, there are two main aspects. The first is the application that may be taken by the telecommunication system user. In this perspective, operational requirements must be developed on the basis of performance criteria for the whole task set by the application. The second aspect is the communicating entity for which operational requirements must be also developed on the basis of performance criteria for those tasks allocated to a communicating entity [5].

The European Union RACE program defined a detailed methodology for service definition in the past [4][5][6][7]. According to this methodology, two stages needed to define telecommunication user services. The first stage describes the goals, the tasks, the users and the performance criteria. The second one provides a decision for the architectural design of a task-based functional communication network that would enable the performance criteria of the application to be met. This task-based network consists of communicating entities and communication channels. The communicating entities must be in specific states in order for them to be possible to execute the tasks. The communication channels are accompanied by specific communication tasks undertaken on each channel. The communication channels include static properties, such as the media quantity and quality, and dynamic properties, such as the user control facilities for channel control and/or communicating entity control. The channel characteristics form recommendations needed to enable the application communication tasks to be completed. In the following, we present the necessary service definitions for the application service level and the telecommunication service level.

3.1. The Application Service Level

An ***Application*** consists of goals, tasks, users and performance criteria and can be defined at various levels of abstraction and extent (i.e. incl. e-shopping, e-learning). In fact, an application includes usage parameters and variables of the *Reference User Services (RUS).* The goal is a categorization of specific services into a small set of generic services. In other words, a method of describing application level services as consisting of one or more building blocks (task level services) is adopted [5].

Main attributes of an application concern the application context, the content, the frequency of use, the number of tasks, the parties involved at the various sites and organizations, the level of cooperation, etc. At the task-level of an application, there is a task taxonomy of generic user services that includes nine task types which come up from the combination of three communications types: *human to human, human to machine* and *machine to machine*, with three basic tasks: *conversation, send* and *retrieve.* These types are the application level generic user services that indicatively include conferencing, distribution, retrieval, conferencing and distribution, etc. The level of cooperation between users is also another important parameter that must be defined [5].

Information is tied together with the user services. Thus, a number of information usage context parameters and variables must be also defined such as the number of different source information types, the amount of information communicated, the type of information transfer required *(audio, still image, moving image, data)*, the content of information communicated, the minimal and ideal quality of information type required and the required transfer rate [5].

As far as the communicating entities are concerned, a number of relevant context parameters and variables must be also defined such as the number and type *(human/hardware/software)* of the communicating, the type of communication *(one-to-one, one-to-many, many-to-one, many-to-many)*, the number of user locations, the distance between locations *(same site, metropolitan, country wide, international)*, the mobility required within or between tasks, the task execution duration and time, the task frequency, the information transfer, the max "waiting time" per task and the call priority [5].

Inter-dependencies between tasks include also important variables such as the desired information processing *(real time, transfer rate)*, the dependencies on the availability of other communicating entities, the required output of information *(transient or permanent, format and content)*, the time control of information delivery, etc. [5].

From the user point of view, users expect telecommunication services of specific quality according to a *Service Level Agreement* and pre-defined *Application Performance Parameters*. These parameters usually imply *reliability*, *availability, responsiveness* and *security*. Security concerns user identification and authentication, communication confidentially and data integrity, anonymity, environmental privacy and repudiation control, etc. Of course, cost to customer and user, as well as usability costs must be also defined [5].

3.2. The Telecommunication Service Level

At this level, static and dynamic service characteristics must be considered and defined. Both consist of the *Operational Dimensions* of the *Specific User Services* that concern the service relevant infrastructure needed to support the task network of the application layer. The

static service characteristics concern the channel level characteristics, the service components, the terminals, the network characteristics, as well as the nested or supplementary services [5].

The *channel level characteristics* include important operational dimensions such as the customer private network (CPN) Relation *(only CPN, internal and external network, only external network)*, the number of tasks supported by the same channel, the number of networks supporting a channel, the number of calls per task, the location of application functionality *(in the terminal, CPN, public net, host, value added service)*, the type of connection (*bidirectional, send only or receive only)*, the type of communication *(one-to-one, one-to-many, many-to-one or many-to-many)*, the call control *(source, sink)*, the mode of establishment *(on demand, reserved or permanent connection)*, the inter-channel dependencies etc. [5].

The *Service Components* can be audio, moving images, still images, data etc. For each of these service components a number of operational dimensions must be defined such as the format perseveration features, the content of information, the quality level, the transfer rater, the maximum delay, the symmetry of information transfer, the mode of traffic, the type of usage, the requirements by combination with other service components *(lip synchronism, location related synchronism, voice synthesis, subtitles synchronized with images)*, etc. [5]

The *Terminal hardware and software characteristics* include the terminal type, the location of terminal, the number of terminals and screens per user, the software types, the peripheral, input, output, pointing devices, the dialogue style, the specific acoustic, lighting, space requirements, etc. The *Network characteristics* include the minimum transfer rate of the network, the network access, fixed or mobile, immediate or delayed, the transfer medium, etc. The *Nested services* include the services like the directory service, while the *Supplementary services* include services like call forwarding, multiparty call and line hunting [5].

The ***dynamic service characteristics*** concern the control of channel level characteristics, the control of service component characteristics and the control of communicating entity states. The *control of channel level characteristics* includes the service request and payment, the change of partners, including redirection, the change of location, the change of terminal, the change of terminal quality, the change of inter-service integration, etc. The *control of service component characteristics* includes the allocation/de-allocation of service component, the change of quality level, the change of transfer rate, the change of symmetry, etc. The *control of communicating entity states* includes the nested service control, the supplementary service control, the control of other communicating entities (e.g. VAS provider, database, terminal), etc. [5].

3.3. A Service Definition Methodology

In this section, we briefly present a proposed service definition method by the RACE program. For each value of a dominant usage context parameter a *Generic User Service (GUS)* is proposed. A GUS is an attempt to organize the *Specific User Services* into a small number of concepts relevant to infrastructural implementation. For each GUS task type (there are nine – see sec.3.1) a family of Specific User Services is produced. [4]. A service, which can be decomposed into more than one task level GUS's, is termed as an application level GUS. Application level GUS's form a conceptual hierarchy since an application level GUS (e.g. store-and-forward) may be a component of a larger application such as medical diagnosis. Thus, the methodology includes an application level *Generic User Service (GUS)* (e.g. medical diagnosis), which implies descriptions for an *application* (e.g. oncological diagnosis), the *application users* (e.g. doctors, patients) and the *performance parameters* (e.g. high reliability, low cost). According to specific rules, task level GUS are selected (e.g. GUS Human to Human conversational and GUS Human to Machine Receive) which are further decomposed in main *building blocks* which can be *main services* (e.g. doctor to patient conversational service) and/or *nested services* (e.g. directory service or file retrieval service) [5]. Finally, for the selected building blocks, *if-then rules* lead to the Operational Dimensions, which are defined according to the following steps:

1. Analysis of task usage context.
2. Analysis of users' goal tasks, breakdown of the major goals into a set of sub-goals and description of the objects that are transformed or created in achieving these goals.
3. Analysis of the required enabling states for the users' goal tasks, defining the system state, the required user's knowledge and processing skills, the service relationships, and the object states.
4. Analysis of the required enabling tasks in terms of *Users, Service relationships* and *Objects* in order for the users to reach the appropriate enabling states for their goal tasks [4].

Figure 1 roughly depicts the methodology presented above:

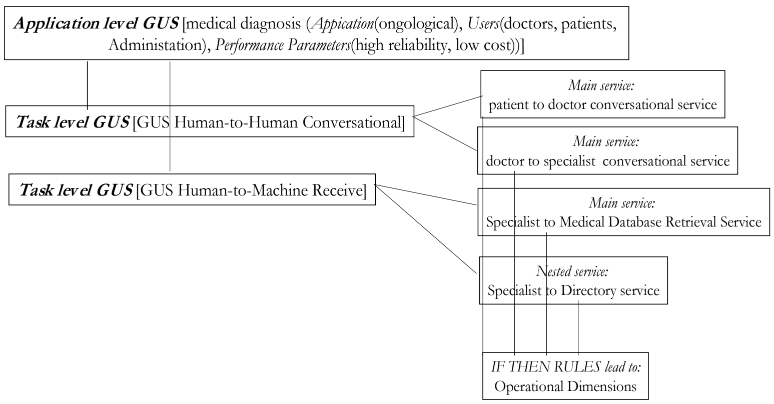

Figure 1: The service definition methodology [5]

4. Service Provision

The service provision usually starts from a business perspective with a service pre-design which provides a rough picture of the service provision concerning both service infrastructure and service distribution to the final customers via specific communication channels. The service pre-design must include the user roles that are involved in the whole service operation, including the service customers, the trusted third parties, the operators, the interface entities (human or machine), etc. This picture must provide –among other things- the initial *modular (molecular) structure* of the service, the service business logic, the workflow processes, the service deployment and operation supporting components, etc.

From the service strategic development point of view, some important issues must first be taken into account. Specifically, such issues are the well understanding of the new service concept, the evaluation of the selected service with respect to the user benefits, the feasibility of the service implementation, the technical and operational testing requirements, the service deployment, etc. Furthermore, the service provider is not obliged to offer capabilities to the user that are not required and dictated by its business plan or business strategy. For example, it is not necessary to offer bandwidth above the required user expectations except marketing purposes. Thus, it is very important for the service provider to understand the customer (service user) requirements, the offerings of related or substitute services, the required service performance, the quality specifications and finally the experience gained from the service operation and use. The latter is a very crucial factor of success and it is influenced by the service customization and personalization, the standards used in service provision, the alternative substitutes in use of the service, the other co-existing service users, etc.

A *service model* must be used in order to explain the above presented factors *(direct and indirect)* that affect the inter-activity between the user and the service, as well as the gained experience from the service use. A generic scheme of a service model includes four main system components:

1. The service user (human or machine)
2. The correlated (inter-dependent) Service Users (Human or Machine) that may influence the experience an ordinary service user may have from the service.
3. The Service Front Office Interface, and
4. The Service Back Office.

These four main components are depicted in the figure 2 that represents the *Servuction* model known from the services marketing area [8][9][10].

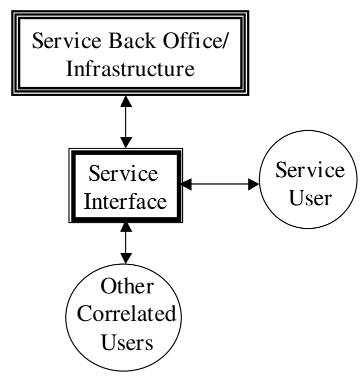

Figure 2: Outline of the *Servuction* Model [9]

A service consists of several parts and relationships, such as:

1. The core service, which covers the basic service business and operational logic. The workflow processes of the core service must follow the *service blueprints*
2. The supplementary services on the core service for the maximum user benefit and satisfaction.
3. The interoperability with external services through appropriate gateways for satisfying the supplementary requirements.

Provisioning a new service, we have first to clarify if the new service is totally new or whether it is based on an existing service. The latter must be taken as a service extension providing specific differentiated experience to the users in comparison with the existing services. The experience gained from an extended service may be quite different from a quite similar service which however is based on a new technology. Thus, it depends on the strategy followed by the service provider which approach will be adopted for the new service development. We note that the service provider must have developed a specific strategic plan in order to examine the pros and the cons of each approach. For example, what is better for a mobile telephony provider, to evolve its GSM services of second generation networks or to adopt and implement and new service based on a third generation network UMTS? In this direction, Service Oriented Architectures can play an important role and contribute to the answering of such questions because they provide the means for flexible infrastructure upgrade, reconfiguration or replacement in a cost benefit and performance effectiveness way.

5. Service Management

Service management is a core system functionality that must accompany service deployment and establishment. An efficient management can only assure the effectiveness of the service operation. There are several distinguished approaches on the area, such as the Telecommunications Information Network Architecture (T)INA, the Telecommunication Management Network (TMN) and the ISO/OSI System Management [11][12][13][14]. Hereafter, we briefly present the TMN as an efficient framework that defines a model of logical layers. These are management layers of specific functionality, such as the *Network Element Layer* (NEL), *Element-Management Layer* (EML), the *Network-Management Layer* (NML), the *Service-Management Layer* (SML), and the *Business-Management Layer* (BML) [11]. The figure 3 depicts the logical layers of TMN and its management interfaces Q3 and X.

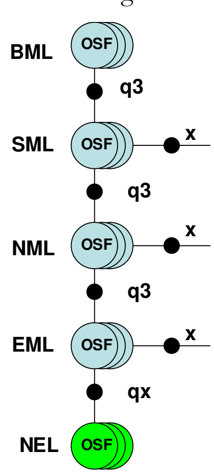

Figure 3: TMN-LLA Functional Hierarchy [11]

Each layer provides with management related services the layer above it. Following a bottom-up approach, we have the following:

1. the NEL that represents the manageability of the individual Network Elements. At this layer, the Q-adapter that implements TMN to non–TMN information adaptation is located.
2. the EML which manages network elements through element managers or Operations Systems (OS's). An element manager is usually engaged with the management of a subset of NE's. An OS represents its management information from a subset of the NEs to an OS of the NML through the Q3 interface. The latter concerns the communication between TMN devices.
3. the NML that provides global view of the network. This view is structured based on the information provided by the EML. NML can manage both individual NE's and groups of NE's. Thus, NML manages the entire network activities and provide the necessary responses to the requests send by the SML. NML OS's and SML OS's communicate via the Q3 interface.
4. the SML uses the management information provided by NML in order to manage services agreed to be provided to the service users/customers. Management functionality, such as quality of service, billing, security, etc. must be supported at this layer. Interoperability with other service providers or administrative domains is also possible. SML OS's communicate with the SML OS's of other administrative domains via the X interface which links the TMN devices of one TMN with the TMN devices of other TMN's. SML OS' communicate with the BML OS's via the Q3 interface.
5. the BML layer concerns business related decisions, such as business goals and planning, business-level agreements, etc. [11].

Typically, components at any level of the TMN hierarchy are modeled as managed objects that must offer both management and user functionality interfaces. Management interfaces comply in various management protocols. Such distinguished examples are CMIP and SNMPv3. The managed objects can be grouped into management domains in order for managers to be possible to enforce common policies on them. The managed objects can be described based on the Guidelines for the Definition of Managed Objects (GDMO) of ISO and the abstract syntax notation language ASN.1 [15][16].

In the following tables 1, 2, 3, we provide an indicative set of managed objects of an IN-based telecommunication network service, including managed objects for the telecommunication network, the intelligent network, a generic class service. The attributes of the managed objects are self-explanatory.

(i) for the *Telecommunications Network:*

Table 1: Managed Telecom Network Objects

Switch ListofInputPoints ListofOutputPoints CurrentInputPoint CurrentOutputPoint	*Adaptor* LineTerminationName LineTerminationList LineName PathTerminationList PathName
Router RoutingTable BufferSize PDUProcessingCapacity NofDeliverdPDU NofReceivedPDU NofWaitingPDU AdjacentRouters	*Link* MaximumBandwidth AvailableBandwidth TerminationPoint_1 TerminationPoint_2

(ii) for the *Intelligent Network:*

Table 2: Managed Intelligent Network Objects

SCP ProvidedService AssociatedSSPs ServiceTime NofServiceReques t	*SSP* AssociatedSCPs AssociatedSignallingNetwork AssociatedInfoTransferAgents AuthorizedTransferSystemUser s

(iii) for the *Service:*

Table 3: Managed Service Objects

Service ServiceType NetworkIDs TariffRules Features ControlElement Components	*ServiceComponent* Type ChargingMode Security Quality Volume AcceptableDelay
SubscriberService SubscriberNames SubscriberProfiles QoSConstraints ConnectionAccessPoints DirectoryInfo	

6. Service Oriented Architectures

6.1. Background

The key ideas of SOA concern the coordination and the composition of Services. SOA's are a set of software components that can be invoked for the success of a service task goal. SOA's must support service interface definition, publication and discovery. Usually, services are loosely coupled and support stateless connections. The latter means that they are waiting in an idle mode until they receive a request. In addition, SOA's due to their (large scale) distributed nature are not under a single ownership or control. SOA's promote the concepts of process workflow, activity coordination, orchestration, and distributed control [17]. The effectiveness of SOA's is achieved by the flexible component-based layered structure. The abstract view of a SOA consists of layers where at the bottom the physical layer, the protocols and the relevant processing stand. Above it, the functional layers stand which are needed for implementing and configuring services, while one layer above the service provision is provided. At the top, the business logic is supported by an integrated distributed service application which serves the business processes using the underlying infrastructure [1].

6.2. XML/SOAP: Communication and Data Exchanging

Typically, a service application stands on top of a service transport layer that uses XML-based enveloping and message formats via Simple Object Access Protocol (SOAP). XML (Extensible Markup Language) is the lingua franca for data exchange. XML describes structured data in plain text. Due to this fact, XML is appropriate for exchanging data between business processes, back-office and legacy systems integration, publishing data and service interfaces, etc. In other words, avoiding proprietary data representations, organizations reduce the dependencies on data formats and improve its operations efficiency. As far as SOAP is concerned, this provides the conventions for service-to-service communications and customer-to-provider interactions. Thus, service providers putting XML/SOAP on top of their platforms adapt their current architectures and take the above advantages.

6.3. SOA Architectures and related Issues

The first implementations of service-oriented architectures were based on CORBA. Even with

CORBA, the implementation was still so tightly coupled to the application. In other words, it wasn't so efficient to develop dynamically value-added services. Other models would be developed recognizing the need for flexible object reusability and service configuration. An example of such a model is the TMF that defines the frame, as well as the mechanisms for web-based services within an SOA [18]. Hereafter, we provide a discussion on Service Oriented Architectures based on Web Services, while figure 3 provides a general scheme of the WS-based Service Oriented Architecture.

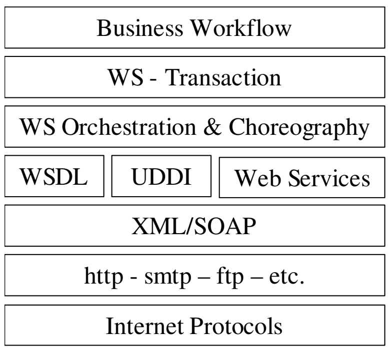

Figure 3: The WS-based SOA approach

As mentioned previously, service applications are created on demand from smaller services that stand on a generic service modules sub-layer, and are integrated by a service integration sub-layer. This process takes place on demand and dynamically as needed. The service integration sub-layer must embed appropriate technology to support integrated business logic construction [19]. This model is provisioned from WS-based architectures where at the bottom line the Network and the XML-based messaging via SOAP stand. One layer above, we have the service description, as well as non-functional descriptions e.g. related to the quality. The service contracts *(service interface descriptions)* are necessary to automate service publishing and discovery via lookup services. [19]. The *Web Service Definition Language* (*WSDL)* describes how to access a WS, while the *Universal Description, Discovery and Integration (UDDI)* provides a registry which can be used by the clients to discover a registered service. Discovering services based on their functionality and/or other characteristics –such as quality- is an important factor for an efficient Web Services operation. In fact, what is put in a registry is the part of a WS interface (e.g. API) which its creator wants to make public in order to be used by other services or clients on a demand basis [20].

Above this layer, *choreography* and *orchestration* stand, which are relatively new approaches for coordination and collaboration of several Web Services. The *integration of WS* is a major objective in order to support flexible configuration of services mix. This layer is also very important for the standardization of cross-organizational business processes [21]. But, WS orchestration has some limitations, such as the dynamic business formation and effective selection of services, the flexible cooperation strategies and the semantic and ontology problems. To cope with such limitations the exploitation of the agents' proactive interaction capabilities can contribute to enhance WS behaviours [23]. In this direction, multi-agent platforms can be helpful and promote applications that can dynamically change their functionality and form new applications at runtime [24].

Above the layer of WS coordination, *WS-Transaction Management* for transactions between WS must be implemented, as well as *business process and workflow*. *WS-Transaction* and *BPEL4WS* specifications provide correspondingly guidelines for the implementation of these layers [22].

The reader can find more on Web Services based SOA standardization efforts in [26].

7. Examples of Service Oriented Architectures

A SOA example is the ***SOCAM (Service-Oriented Context-Aware Middleware)*** platform that supports context-aware services building in *pervasive computing*. SOCAM is a distributed middleware that "converts various physical spaces where contexts are acquired into a semantic space where contexts can be easily shared and accessed by context aware services" [25]. SOCAM is based on an ontology-based context model and consists of the *Context providers, the Context interpreter* for logic reasoning services, the *Context database* for ontology storage, the *Context-aware services* that make use of different level of contexts, and the *Service locating service* [25][27]. Figure 4 depicts the layered structure of SOCAM.

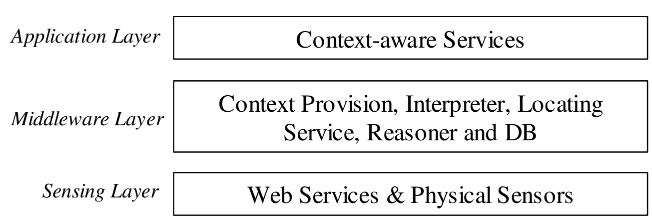

Figure 4: The SOCAM layers

Another distinguished example is the ***UMTS OSA (Open Services Architecture)*** which is a 3GPP standard. The main idea of UMTS OSA is to open the network for application development. In this direction, UMTS OSA provides a number of standardized application programming interfaces that can be used for the networked services implementation [28]. Supplementary, UMTS OSA allows external service providers to integrate their services into the core network. Figure 5 depicts the UMTS OSA layered architecture.

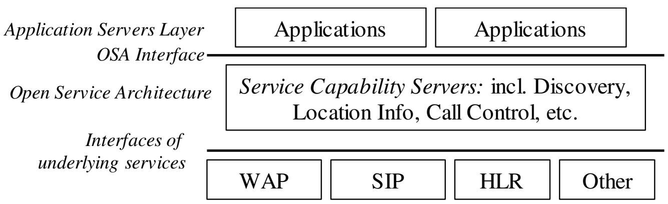

Figure 5: The UMTS OSA layers

The core of UMTS OSA architecture consists of the Service Capability Servers *(SCS)* that represent one or several OSA features. Applications that are hosted by a number of Application Servers access the networks and terminate at the customer site by using the offered OSA API's. The goal is the access of operators to the various network facilities by abstracting the underlying network access. While applications combine different network capabilities, fixed/mobile/IP migration and convergence is achieved [31]. OSA supports management functionality through a number of *Service Management Servers (SMS)*. Such functionality includes security, billing, directory services, fault tolerance, performance assurance, etc.

Another important example is the ***Open Grid Services Architecture (OGSA)*** - specified by the Open Grid Services Infrastructure Working Group (WG) of the Global Grid Forum (GGF) [29]- which defines a core set of capabilities for Grid systems based on Web services technologies and on resource management. In OGSA, grid resources (soft or hard) are modeled through web services, while grid middleware is composed of a number of core services that interact with each other [32]. At the highest layer of OGSA stand the applications that make use of grid capabilities. The latter mainly concern the integration of grids of data of any form and grids of distributed computational systems, resulting in accelerated application performance and open system interconnection. OGSA framework exploits the potentiality of WS within grid computing.

The Open Grid Services Infrastructure (OGSI) defines and builds Grid services on top of Web services platforms. In other words, OGSI employs WSDL to specify standard interfaces and interaction between Grid resources. In addition, OGSI provides additional capabilities such as state management, discovery services, notification services, life cycle etc.

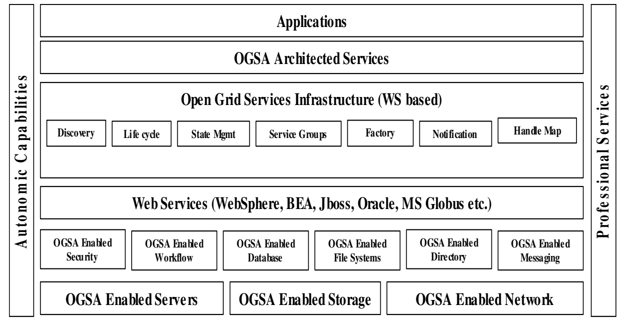

Figure 6: The OGSA Framework

8. Conclusions

Market competition and the introduction of new technologies push service providers to provide new or augmented services to their customers quickly and with low cost. Service Oriented Architectures, the relevant new approach in system and service design, have proved very effective for such a purpose. Indeed, SOA among other things provide the appropriate conceptual designs, deployment platforms, usage profiles, service registration, discovery and reusability, flexible management and support the further evolution of process orientation in business environments [1][30]. For these reasons, service providers are turning to a new networking paradigm. Networks gradually are evolved to SOA improving in this way business productivity, agility and responsiveness to customer demands and expectations. From the technological point of view, web services and component based software development tend to become a de facto standard, because they support the distributed application development in a by far more easy way than the traditional approaches. Platforms such as Microsoft's .Net and Sun's J2EE, along with XML-based enveloping and message formats become wide spread in the world of the new applications development.

This paper provided an integrated overview of all the main service related issues, such as definition, provision, deployment, management, object-orientation and architectures for both the telecommunication network providers and the Value Added Service providers.

References

[1] Margaria T., Stffen B. Service Enginnering: Linking Business and IT, IEEE Computer Magazine, pp.45-55, Oct 2006

[2] Mitropoulos et al. Advanced Telecom Services in Multimedia Tourism Applications. Proceeding of Melecon 1994 IEEE Conference, Antalya, 1994

[3] Licciardi et at. An Architecture for IN-internet hybrid services, Computer Networks, Vol.35, No.5, pp.537-549, 2001

[4] RACE Project R1067. Usability Engineering Methods for IBC Services, Dec 1992

[5] RACE Project R1077, Service Definition Methods for IBC Services, IBC URM, July 1991

[6] RACE Project 1077. RACE Usage research Review, IBC URM, Dec. 1991

[7] Byerley, P. May, J., Whitefield, A. & Denley, I. (1991). The Enabling States Approach: designing usable telecommunications service. Journal of Selected Areas in Communications, , Vol.9, No.4, pp.524-530 , IEEE, May, 1991

[8] Holmlid S. Service Design methods and UCD practice, INTERACT 05-Workshop: User Involvement in e-Government development projects, Oslo, 2005

[9] Langeard E., Bateson J., et al. Marketing of Services: New Insights from Consumers and Managers, Report No. 81-104, Cambridge, MA: Marketing Services Institute, 1981

[10] Zeithaml V., Bitner M. Services Marketing, McGraw Internation Edition, N.York, 1996

[11] IEC site, *http://www.iec.org/online/tutorials/tmn/*

[12] ISO/IEC, Systems Management Overview, Standard: ISO/IEC 10040.

[13] Journal of Network and System Management, special issue on TINA, Vol.5, No.4, Dec. 1997

[14] R.Gupta, and P.Cook, Technical Assessment of (T)INA-TMN-OSI Technology for Service Management Applications, Proc. Network Operations and Management Symposium, *IEEE/IFIP,* pp.877-887, 1994

[15] ISO/IEC IS 10165-4: Information Technology – Open Systems Interconnection – Structure of Management information. Part 4: Guidelines for the Definition of Managed Objects, 1992

[16] ISO/IEC 18824: Information Technology – Open Systems Interconnection – Specification of Abstract Notation One (ASN.1), 1990.

[17] Bocchi L.et al. On the Impact of Formal Methods in the SOA, Electronic Notes in Theoritical Computer Science, Vol.160, pp.113-126, Elsevier Science Publishers, 2006

[18] TeleManagement Forum, *http://www.tmforum.org/*

[19] Turner M, Budgen D., Brereton P. (2003). Turning Software into a Service. IEEE Computer Magazine, pp.38-44 October 2003.

[20] Menasce D. et al. QoS management in service-oriented architectures, the International Journal of Performance Evaluation, *Article in Press*, Elsevier Science Publishers, 2006.

[21] Peltz C. Web Services Orchestration and Choreography, IEEE Computer Magazine, pp.46-52, Oct 2003

[22] W3, *http://www.w3.org/TR/ws-chor-model/*

[23] Weiming S. et al. An agent-based service-oriented integration architecture for collaborative intelligent manufacturing. Robotics and Computer-Integrated Manufacturing, Article in Press, Eslevier Science Publishers, 2006

[24] Tsai W., (2006). A service-oriented modelling and simulation framework for rapid development of distributed applications. Simulation Modelling, Practice and Theory Vol.14, pp.725-739. Elsevier Science Publishers, 2006

[25] A Service-Oriented Context-Aware Middleware, *http://www.comp.nus.edu.sg/~gutao/gutao_NUS/SOCAM.htm*

[26] Hamid R., Motahari N., Boualem B. et al. Web Services Interoperability Specifications, IEEE Computer Magazine, pp.24-32, May 2006.

[27] Tao G. et al. A service-oriented middleware for building context-aware services. Journal of Network and Computer Applications, Vol.28, pp.1-18, 2005.

[28] *http://www.w3.org/2001/03/WSWS-popa/paper59*

[29] Globus, *http://www.globus.org/ogsa/*

[30] Tselikis C., Mitropoulos S., Douligeris C. An Evaluation of the Middleware's Impact on the Performance of Object Oriented Distributed Systems. Journal of Systems and Software, Article in Press, Elsevier Science Publisher, 2006

[31] Parlay and UMTS Open Service Architecture, www.tina.com/conference/proceedings/

[32] Corcho O. et al. An overview of S-OGSA : A Reference Semantic Grid Architecture, Journal of Web Semantics, Vol.4, pp.102-115, Elsevier Science Publishers, 2006