New Services Architecture - Interface Design

Mehdi Khorasani† and Dr. Lionel Sacks‡

† Lucent Technologies, ‡ UCL, Dept. of E&EE

Abstract: The new services architecture that is in the process of standardisation is emerging the data and telecom trends into a single stream. Triggered within the telecom industry Parlay, OSA suggests an API-based approach while the Web Services technologies recommend a WWW based service model. SIP provides an IT based service control and JAIN tries to combine the advantages of Java technology with the API approach.

The realisation of the new services architecture is a major challenge within the industry. Since the networks are vastly distributed, the engineering of the architecture needs to be based on the distributed computing technologies.

This paper studies the impact of interface design on the overall integrity of the new services architecture. The study contributes to a wider research on defining a new integrity model.

1. New Services Architecture - Overview

The new data and telecommunication services architecture that is converging has notable characteristics that distinguishes it from the traditional service delivery model. It is now established that the trend has been moved away from defining individual services to standardising service capabilities. Some of the emerged leading service architectures in the industry are introduced below1.

In telecommunications and within the context of the Next Generation Mobile Networks (henceforth referred to as 3G networks), three service mediation technologies have been developed, i.e. Open Service Access (OSA), SIP Application Server and CAMEL [1].

OSA /Parlay is an open network API technology that allows application and service providers to define more advanced services using various networks' capabilities in a secure (for both application providers and network operators) and open way. 3GPP that has been responsible for producing technical specifications for 3G networks' architecture (in particular UMTS) in its Release 5 has introduced an IP backbone subsystem, IP Multimedia Subsystem (IMS). "The IMS introduces a multimedia call control model [2] that is based on the Session Initiation Protocol, SIP, defined by the IETF [3]"[4]. On the other hand the 3G networks expect to expose the existing networks' capabilities and provide the end users with the traditional IN services. To address this the CAMEL (the application of Intelligent Network concept to the mobile systems) service architecture has been further developed by 3GPP.

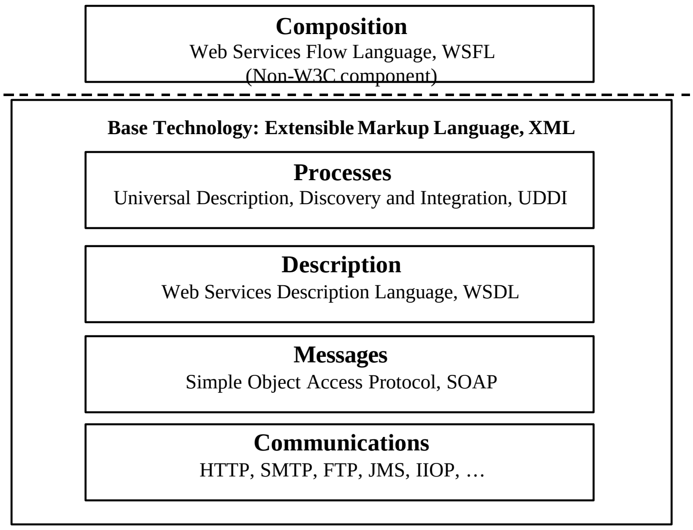
In the data communications and the Internet domain two main shifts can be identified, i.e. (1) the introduction of voice over packet switching that resulted in developing new real time service control mechanisms such as H.323 [x] and SIP, and (2) the development of Web Services technologies as a new service architecture.

While the standardised H.323 has been mainly used for providing the voice service over IP, the de-facto SIP technology has been widely accepted and also further enhanced as an efficient multimedia session control means.

On the other hand the Internet as a compelling, globally available communication medium has triggered the development of the Web Services technologies, which have been standardised by World Wide Web Consortium (W3C). Web Services are a collection of complimentary interface technologies that enable business entities and applications to openly intercommunicate with each other over the network. Web service architecture involves many layered and interrelated technologies. There are many ways to visualise these technologies, just as there are many ways to build and use Web services [5]. The following picture provides one illustration of the Web Services stack families.

---

[1] For the purpose of briefness JAIN has not been discussed in this section.



Web Services allow the exchange of XML data and its corresponding semantics between software agents, i.e. services. SOAP is an RPC-style messaging protocol for the communication of agents over the 'communications' layer. Services and their connections and messages can be described using WSDL. UDDI is a set of protocols that can be used to register services in a public directory, to look up and discover services and other business processes. WSFL, an IBM proposed component, is an XML-based language for the description of Web Services compositions. WSFL can be used to describe the business logic in terms of assembling various web services into an end-to-end business process.

The new services architecture paradigm allows the network operators to provide the end users with the existing and a wide range of new services developed by third party application and service providers. A big number of the new services are aiming at allowing the end users to enhance their experience when accessing the network services.

2. Engineering of the New Services Architecture

The engineering of the new services architecture is a complex challenge mainly due to the *distributed* nature of the *communication* networks and also because of the *layered/API* character of the new architectures. These have significant consequences on the engineering of the new systems as discussed below.

Entities within a network communicate with each other using communication protocol stacks. Communication protocols and their implementation is a coherent part of the system engineering. Protocol violation is still an unresolved issue in that area.

The systems are distributed. This implies that entities, which are not working on the same processing environment, need to work together to implement some intended functionality/service. Some of the key issues to carefully consider are unpredictable latency of remote access (i.e. timeout), concurrency and possibility of partial failure. Today, there are a number of advanced CBSE-technology[2] based [6] and ORB products available on the market, e.g. COM/DCOM and many CORBA implementations.

On the other hand the layering and API nature of the new services architecture has its own consequences. The advantage of abstraction of underlying network complexities and capabilities in the API-based architectures is to some degree counteracted by the loss of detail [7]. This problem is somehow similar to protocol violation, i.e. how can an application that is running on top of an API or a particular layer in a protocol stack access information that is residing within the lower layer and has not been presented or abstracted through the used interface? A number of ideas have been developed to overcome this problem. The most straightforward and efficient way of avoiding this problem is to choose the right layer for an application from the outset. However it is not always possible, e.g. in the case of Parlay/OSA APIs the application layer (on top of the APIs) and the network layer (underneath the APIs) are residing within two separate security domains. In this example one way of solving the problem is to extend the APIs to abstract the required capabilities. If it is done through standardisation it will add to the complexity of the APIs that in turn will have its own consequences. If it is done

[2] Component Based Software Engineering

through informal mutual agreements it causes ‘openness violation’. Can the industry develop the concept of ‘Intelligent Interfaces’? The idea will be expanded in the future papers.

Another issue with the API approach is the fact that at architectural level, the idea is to design interfaces and APIs as generic, implementation and platform independent as possible [7]. A question is what is the best choice for underlying transport technology? In the case of Parlay/OSA APIs, up to Parlay version 3.1 and 3GPP R4 the only supported realisation technology was OMG IDL that suggests the use of CORBA as a transport technology. In the later releases XML/Web Services technologies are being considered as technical realisation candidates.

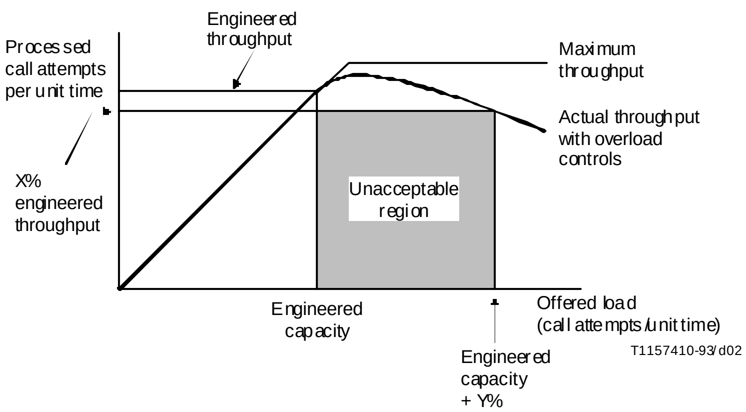
Interoperability is the next challenge. There is no single network in the globe provided by a sole vendor. In the circuit switching the interoperability was achieved by means of protocol standardisation and the use of Abstract Syntax Notation No 1, ASN.1, as an on-the-wire platform independent encoding/decoding mechanism. In the new heterogeneous circuit-packet switching environment with many different realisation options it is yet an open issue that needs to be addressed in a satisfactory manner. An example is the interoperability of an OSA application client implemented in JAIN SPA from one vendor with an OSA Framework implemented in XML/SOAP or on a CORBA platform. Even the issue of interoperability of the entities from different vendors but based on the same transport technology has not been resolved yet. In the case of Parlay/OSA APIs many non-functional aspects of the architecture has been left outside the scope of the specifications.

Software engineering is another side of the realisation. It is directly linked to the choice of the service architecture and the underlying communication technology. With the exception of Java technology the choice of the software technology and programming language does still have a direct impact on the selection of the underlying platform and transport technology and this in turn impacts the interoperability of entities from multi-vendors.

3. Interface Design and its impact on the Overall Integrity

In the traditional well-established circuit-switching world and prior to the shift to the new paradigm the following features secured the integrity of the networks to some degree:

The standardisation of the circuit and non-circuit related signalling protocols e.g. SS7, ISDN, ISUP, INAP, etc was maintained by international and continental bodies such as ITU-T, ETSI or continental-wide private agencies such as former Bell System, Bellcore. These bodies or agencies defined the entire large complex systems. A huge amount of effort was spent on securing end-to-end interworking when developing new technologies. Examples of this effort are “ITU-T Q.706 Recommendation - Signalling System no. 7 MTP Signalling Performance” where they recommend 20% link occupancy, “ITU-T Q.543 Recommendation – Digital Exchange Performance Design Objectives“ that defines, amongst others, the offered vs. processed load (figure below), and “Bellcore - LATA switching systems generic requirements” which recommends the availability goal of 99.81% that amounts to 10 minutes of outage per year. The old process was slow but at least there were virtual bodies responsible for end-to-end overall integrity to some degree.



ITU-T Q.543

Throughput performance with overload control activated

In this environment a number of integrity frames have been developed. In most cases a number of system attributes which impact the integrity of the system (i.e. the entity providing a service/function) has been defined or identified, for example Reliability, Resilience, Robustness, Availability, Performance, Scalability, Modularity, Coupling, Complexity, Safety, Risk, etc. These attributes usually have been contemplated when designing a system, deploying it and during its operation.

Despite all these efforts the industry experienced many network failures, e.g. well-documented integrity breaches that occurred between 1988 and 1991 in the US [8].

The new way of thinking has influenced the working methods in standardisation bodies with its own pros and cons. Bodies such as 3GPP, 3GPP2, IETF, W3C, Parlay Group are producing hundreds and hundreds of open standards for components and partial systems but nobody seems to be in charge of the end-to-end integrity of a complex network. At the most, the standardisation bodies are working on interworking between a limited numbers of technologies.

If the new environment is moving towards more openness, more players, more components and forum-type of standardisation, the mentioned integrity approaches become even less applicable to the new systems.

A new integrity frame is needed that should inherit the main characteristics of the new environment. If the traditional trend was to define the integrity from top to bottom and from entirety to entity, the new approach needs to reverse it. If the new integrity frame consider a system as a life organ (Hyrda) with self-sufficient, self-organising, de-coupled, interconnected components, then *the overall degree of integrity* of a system can be defined as the ability of its units to operate to their functions and performance and to co-operate, which will contribute to the overall system specification.

The characteristics of this approach can be summarised as:

- *Dynamic levelling Integrity* as oppose to *static Integrity*
- Entity to entirety
- Formulate Integrity at architectural and design stage
- Designing systems with the smallest units as possible
    - Minimise the integrity breach propagation by localising them at ‘cell’ levels
    - Conform to modular and layered architecture
    - Conform to open architectures with open standard interfaces
    - Provide a way to design complex systems as an integral collection of components with simplified functions (like RISC processors)
    - Components re-adjust themselves
    - Challenge: Minimising the impact of component failure, increases the components’ interaction → Intelligent Interfaces

The frame will be further developed and explained in the future works.

References

[1] Standardised Service Capabilities: The need for standardisation illustrated by the development of the Presence Service, *P.M. Carpenter*, *M.L.F. Grech*, IEE 3G2002 Conference, May 2002.

[2] 3rd Generation Partnership Project, Technical Specification Group Core Network, “IP Multimedia (IM) Session Handling; IP Multimedia (IM) call model; Stage 2 (Release 5)”, 3GPP TS 23.218

[3] Internet Engineering Task Force (IETF), RFC 3261: “SIP: Session Initiated Protocol”

[4] Service Control for NextGen Applications in Wireless IP Multimedia Networks, Lynell E, et al, Bell Labs Technical Journal, Volume 8, No 1, 2003.

[5] Web Services Architecture, W3C Working Draft 8, August 2003, www.w3.org/TR/2003/WD-ws-arch-20030808/

[6] Component-based Integrated Systems Development: A Model for the Emerging Procurement-centric Approach to Software Development, *V. Tran*, Computer Software and Applications Conference, 1998. COMPSAC '98. Proceedings, 19-21 Aug 98, p 128-135.

[7] A Study of Openness in 3G, M. Khorasani, L. Sacksl, London Communications Symposium 2003, London.

[8] Public network integrity - avoiding a crisis in trust, *J. McDonald*, IEEE Journal on Selected Areas in Communications, vol. 12, issue 1, Jan. 1994.