White Paper:
Delivery of Integrated Services in AlbatrOSS

Authors : Mickael Badar (Atos Origin), Tamara De Vega (Telefonica I+D), Hervé Karp (Atos Origin), Mario Lopez Gallego (Telefonica I+D), Philipe Perez (Atos Origin)

Abstract: *This paper presents some results from AlbatrOSS, a European R&D project partially funded by the European Commission*. You can find more details about AlbatrOSS from the web site: http://www.ist-albatross.org/. The service market is evolving. In the AlbatrOSS project we tried to analyse the changes that this is bringing about and how they will influence the architecture of the systems dedicated to the delivery and operation of services. In the Delivery of Integrated Services sub-trial, one of the three sub-trials developed and validated in the AlbatrOSS project, we present an innovative approach to integrating the need for service composition based on different parties. The main ingredients of this approach are: the definition of service lifecycle phases linked through service models, and the definition of a system architecture implementing NGOSS principles through the use of Web Services technologies.

Keywords: Service Delivery and Management platform, Service lifecycle, Service models, Integrated Services provider, Service aggregation, Service design, Service oriented Architecture, Web Services

Introduction

The telecommunications market is still undergoing radical changes. This is based on different facts and the main one today concerns the convergence between telecommunication and Internet services, pushing an evolution of telecommunication technologies towards information technologies. To analyse such a complex and rapid evolution we need to look at the needs of the different actors and the new value chain which is emerging.

The AlbatrOSS project has considered such an evolution of the market and technologies and has proposed innovative solutions to deliver and operate services in the emerging 3G environment. This paper presents some of the project results focused on the definition of a Service Delivery and Management platform related to a new approach to building services. The starting point is an analysis of the context (see section Context). Based on these requirements, we present the concepts we used or reused to elaborate the global service environment (see section Concepts). Then we present how a service is designed based on modelling techniques and the integration into a service lifecycle, allowing the different phases of creating, deploying, running and operating a service to be followed (see section Service Lifecycle Models). In the last part, we present a proposed architecture for the Service Delivery and Management platform enabling integration of the models defined in the design models (see Service Delivery and OSS Architecture). This architecture has been implemented through one sub-trial (*Delivery of Integrated Services*) of the AlbatrOSS project. A Travel Assistant service has been defined and integrated into a scenario to show the capabilities of the Service Delivery and Management platform and to prove the viability of the approach and concepts. A film of this scenario can be viewed from the AlbatrOSS web site.

Context

The mass market has been the main driver of the 2nd generation mobile market. Mobile operators still compete around this mass market in the 2.5G market. Evolution towards 3G services could bring more visible segmentation of the market, i.e. between mass market and business market.

The real need for the mass-market users is to access a maximum number of services from one terminal, anywhere, for a reasonable price. The needs of business people can be more stringent as soon as business operations start relying on services depending on security levels and specific QoS. This highlights the necessity for SLA definitions in a B2B environment. However, we should not forget that user loyalty is based on the global quality of service. Price is not the only criterion for selecting a mobile operator. Therefore, even the mass market operators and service providers need to consider service level management as a business-related issue.

The right positioning for Mobile operators and global telecom operators is becoming increasingly complex and requires more and more flexibility to give the right answer at the right time.

How to apprehend the evolution of technologies, such as UMTS, WLAN, and how to generate new profits from these changes? Is the competitive strategy to be focused on price? What about QoS? These are the main challenges for Network Service Providers.

One important change to be considered concerns the redefinition of the business models and the positioning of the actors in the evolving service value chain. Models such as those defined by the global telecom operators in Europe (Deutsche Telekom, Telefonica, France Telecom) with different organisations: one for the mobile market and another for the Internet one, could very soon need to be repositioned to sell Mobile Internet services and global package services (including fixed and mobile services). These global telecom operators could take advantage of the large service offer they can provide to their customers. The service value chain needs to integrate new actors such as MVNO and WLAN service providers. All this has obviously a direct influence on the business, but we should not underestimate the radical change in the systems (OSS, BSS, Service Platform) necessary to make things evolve.

There is a common understanding in the operators' community that benefits will not come from the network and connectivity services, but from added-value services as well as content and application services. The competition between service providers is more and more about the range of services they can offer and the speed in bringing new bundled services to market (one for the children, one for the 30-year olds, etc…). Added-value services are quite important in this competition game, including location-based services and presence services. Speed in creating new services has a direct influence on the business perspectives.

The operators always need to offer new services. It is therefore necessary to integrate services provided by Application/Content Service Providers. It is an interesting possibility for network operators to bring to their users unlimited access to content and applications. However, bringing ASPs into the new service value chain is not straightforward. ASPs aim to deliver services that users will want to pay for. Therefore they need to provide quality services that can be accessed by a maximum number of users. Such an environment can be provided to ASPs by the Mobile Service Providers (see Figure 1).
It is quite complicated today to achieve a good mix between technology and market demand. We cannot consider only the one-way process where supply is stimulated and oriented by the demand. There is sometimes the need to sell the technology and therefore to stimulate the demand. This is quite problematic as sometimes the justification for investments in new technologies, such as UMTS, is really hypothetical. We are still looking for the killer application and this is a quite difficult task as it depends on lots of factors, including economic factors. Some people think we will jump directly to the 4th generation, with no stop at 3G. But we could consider that going from 2G to 3G was a too big step for the market and it was necessary to define an intermediary step with 2.5G. Nobody knows the truth for tomorrow.

AlbatrOSS Business Model
The AlbatrOSS project and other IST FP5 projects have investigated the integration of new technologies for the provisioning of new services such as location-based services, context-aware services or presence services as well as a Smart Home Environment. These are based on the integration of different technologies with new terminals (PDA, Wearable computer, …), new network technologies (WLAN, UMTS) and new service platforms (e.g. based on OSA /Parlay).
Based on the context analysis as described above, AlbatrOSS has proposed one business model relying on the service value chain (see Figure 1). The main ingredients of this business model are the roles and the relationships between these roles. Therefore, one focal point in the service delivery to the end users has been defined as the ***Mobile Service Provider (MSP) or Integrated Service Provider (ISP)***. It is in charge of the delivery and management of integrated services for the end users. Integrated Services (or aggregated services) are based on connectivity services (and network services) and content/application services. These services can be provided by third party service providers: network operators for the connectivity services and ASPs for the application/content services. This business model allows an evolution of the Telecom operators towards an integrated services provisioning which will keep customers' loyalty, thanks to the very large offer of value-added services, content/application services and access to services independent of user location and access network.

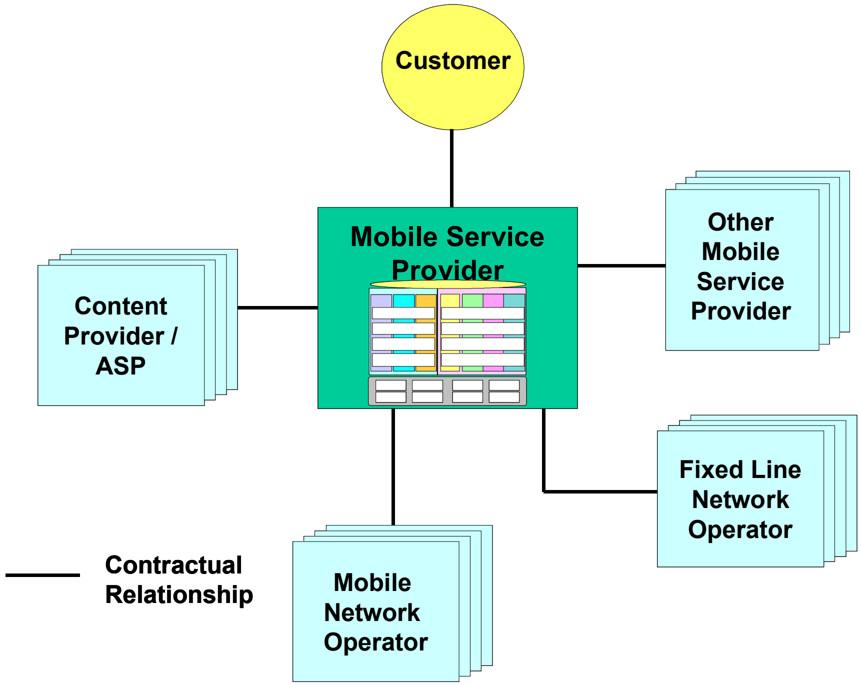

Figure 1: AlbatrOSS Business Model

The role of the MSP is not only to integrate core service components and to provide an integrated view to the end user, but also to integrate management information from third party providers and to react as necessary to assure end-to-end service quality. This evolution brings important changes to the systems in charge of service delivery and operations support (OSS). One main change concerns the evolution from a network-oriented management system (OSS) towards a service-oriented one. We need to consider services separately from the networks, so that we can build services independently of the underlying network technologies. The next change concerns the complex relationships which need to be managed with third party providers and the necessity to provide end-to-end service management.

Based on the provision of new services, such as location-based services or presence services, the number of interactions between service delivery and service management is increasing a lot. The Business Processes and components from both parts would benefit from being built according to the same framework architecture and we therefore propose a Service Delivery and Management platform.

Integration of these mixed requirements in order to provide a solution has been achieved through the definition, in a first step, of concepts.

Concepts

The AlbatrOSS project has been divided into three sub-trials, allowing different aspects of the enabling environment supporting 3G services to be focused on. This paper is presenting only the ***Delivery of Integrated Services*** sub-trial. Each sub-trial defined clear objectives in terms of requirements to be fulfilled. The link between the sub-trials is kept as this work started from a common business analysis (including a common story-board), as well as one state-of-the-art architecture framework (including methodology guidelines). This allowed the project to present different aspects to be considered in the evolution of service and network management systems in a 3G environment. ***The main concepts addressed by the DIS sub-trial are: service aggregation, NGOSS-based Service Delivery and Management platform and Service Lifecycle modelling approach***.

As presented in the section below, AlbatrOSS defined a main role in the business model selected: the ***Integrated Services Provider (ISP)*** or Mobile Service Provider. This role is the focal point for the user to access, but also to subscribe to and to pay for services. We define services provided by the ISP as ***aggregated services***. They integrate content and connectivity services, which can be provided by third party providers. The ISP concludes an SLA with the end user and thus is in charge of the end-to-end service management. Here the main issue is to define how to integrate services provided by Network Operators (connectivity services and other network services) with content/application services provided by ASPs. Moreover, these different services can interact all together to bring added value to services.

One of the concepts developed here is about defining an application level independent of the connectivity level. It means end-user services can be accessed from different access networks. It also means that ISPs can create new services independently of the underlying network technologies. OSA/Parlay specifications [Parlay_Group] are based on this principle. The OSA/Parlay API aims to decouple the application level from the network level. There also exist other options to enable applications to access the connectivity level, such as SIP and the ALL IP vision.

The option we selected, i.e. OSA/Parlay, provides the possibility of controlling the session and of bringing terminal and user information to the application level through the network (location, user status, etc…). For the implementation of the service aggregation concept we would have needed a definition of the management interfaces integrated in the OSA/Parlay API, thus allowing the ISP to assure end-to-end service management. This is not provided, or only partially, by OSA/Parlay, therefore in the DIS sub-trial we developed a specific management interface. One option could be to use OSS/J API [OSS/J] and perhaps define how this could be integrated with the OSA/Parlay Framework. One interesting development on top of the OSA/Parlay API, which is defined as a CORBA IDL interface, has been done by the Parlay X working group [ParlayX]. This group provides Web Service interfaces (WSDL, SOAP) on top of OSA/Parlay. This has been used in the DIS sub-trial. We will see later the impact of a Web Services oriented architecture.

On the other hand, the ISP needs to integrate services provided by ASPs. For this we selected Web Services technologies as it seems the best option regarding interoperability issues in such a context. Web Services also provide service discovery mechanisms (UDDI). These have not been used in the DIS sub-trial, but could be an option for implementing some service discovery processes. It is also interesting to define the ASP as providing services, and not only application or content, to the ISP. The notion of service integrates the supplier/customer relationships. However, even if Web Services arouse a growing interest, we need to solve some issues around security in order to use Web Services between different authority domains. There is also the need for service management interfaces. Some recent initiatives could be investigated such as WSLA [WSLA] as well as the B2B standard ebXML [ebXML]. We also need to consider a precise description of the service flow from WSDL; this is considered by the DAML-S initiative [DAML-S] and has been investigated in our sub-trial.

Inter-ISP relationships, allowing roaming services, have not been investigated in the DIS sub-trial, but this has been done in the other AlbatrOSS sub-trials.

Based on the model we selected, network provider, ISP and ASP domains are separated. This means that we need management systems in each domain, with the ISP management system in charge of end-to-end service management. Therefore, the network provider runs a network management system (OSS) and the ISP a service management one, which could be defined as a service-oriented OSS. This is in line with the orientation of eTOM [eTOM] which evolved from a focus on the network towards a service-centric OSS. We propose going further.

First, the network management platform is separated from the service management platform (the network management platform is in charge of the management of network services). Second, we need to consider the fact that service management processes are increasingly interrelated with service delivery processes. Added-value services such as location-based services or presence services, bring tight coupling between service management functions and functions more related to the service delivery, for example around adaptation of the service to the user context. We propose therefore to bring the complete view of the processes into a common process framework map. This could be defined as an extension to the eTOM process map.

Another point related to the SDMP (Service Delivery and Management Platform) concerns the architecture. The two main constraints we defined for the platform were: to support the possibility of adding SDMP components as plug'n play and to automate the creation and deployment of services. The two points are major objectives of the NGOSS framework. Thus the architecture of the SDMP we prototyped is based on NGOSS architecture principles. This is detailed in the following section and in the AlbatrOSS NGOSS white paper [Alba_NGOSS]. There are two important aspects we added to NGOSS: the service lifecycle modelling approach with the definition of an aggregated service entity in the SID models and the use of Web Services technologies to implement an NGOSS system architecture. The service lifecycle modelling approach is presented in the section "Service Lifecycle Models" and the NGOSS architecture is presented in the section "Service Delivery and OSS Architecture". The SID model adaptation to AlbatrOSS is presented in [Alba_NGOSS].

Service Lifecycle Models

As we saw earlier, the business model in the telecommunications market is evolving towards a service-based relationship. This is true for end-user/ISP relationships but also for relationships between ISP and Third Party Providers. A major need is to put the customer/user in the centre of all processes, i.e. service delivery and management. If you look only at this point, it means a great change to how services should be created, or even what the service creation process is, as we could think about user-specific services.
It is important here to define what our definition of service is. *The service is characterised by all the interactions between a service provider and a customer/user.* Therefore the service is not only the core service (content or application) but the service includes the management aspects (subscription, CRM, Billing) as well as added-value services such as roaming, presence, etc … In such a user-centric approach, part of the service is built when the customer subscribes or even when the user uses the service (we talk about context-aware services).
Another need for the service providers is to be able to offer a bundle of new services to its customers. These new services have to be provided to customers very rapidly because of the competition. This also has a direct impact on the service creation, deployment and operation processes.
The answer to such requirements could be to automate all processes as much as possible and to create reusable service components.
It is important to not consider the different steps of the service lifecycle separately, instead we should give an integrated view of the whole service lifecycle processes. *Usually, service creation, service deployment and service operation were considered as definitely separated, which means running on fully separated platforms. This is incompatible with the strong need to shorten the creation and deployment processes.*
We propose using a modelling approach and therefore model the service, the interactions with the users and the components of the service. The service is to be considered as a puzzle (composed of service components). But the component description (the reusable blocks) is not enough; it is necessary to describe the logic between the components, defined as the service logic.
Below we will briefly present the service lifecycle steps and processes, and then we will focus on the different service models related to the *service design process.*

Service lifecycle processes
The service lifecycle is separated into three main steps:

- ***Create***: Addresses the specification, design and implementation of the different services.
- ***Deploy***: Addresses the service installation and configuration on the SDMP.
- ***Operate***: Addresses the operational aspect of services, which identifies the different interactions between the user and the service, from subscription to billing.

The different steps are represented in the figure below.

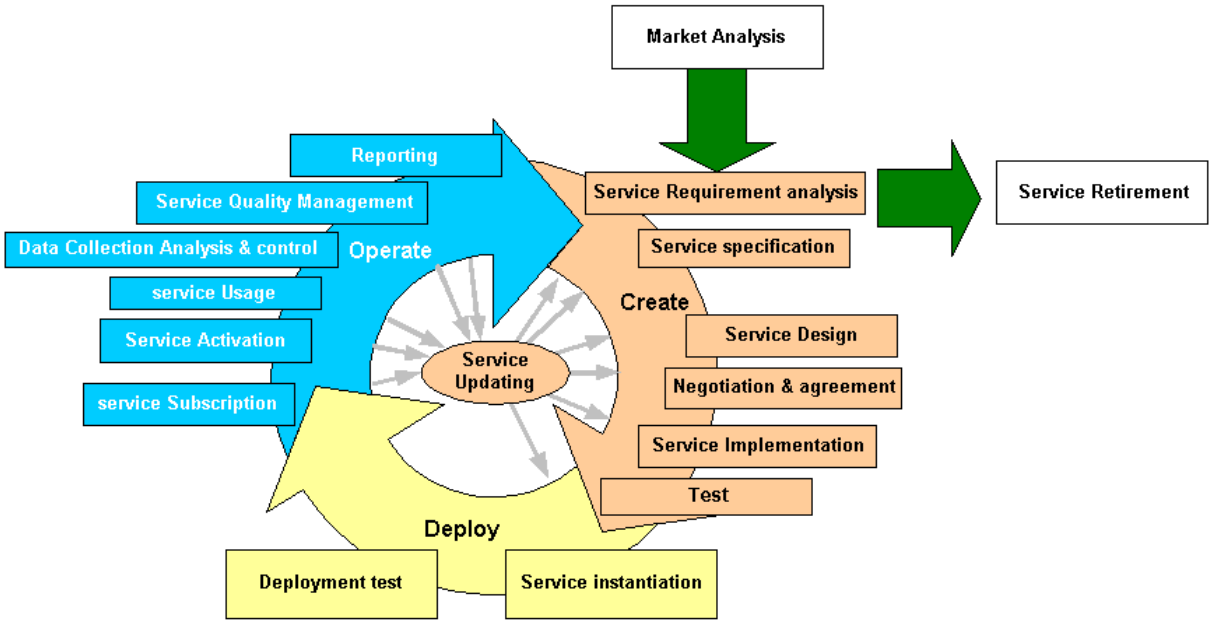

Figure 2: Service lifecycle processes

The Service Update process in the middle of the figure shows that the service can be updated based on a change request which could occur in different operation steps (it could also come from a business decision). On the other hand, the Service Update process can be integrated into different steps of the service lifecycle depending on the nature of the change. In short, it means that the service does not need to be re-created to integrate change, but thanks to the modelling approach it can be rapidly adapted.

The table below shows the mapping between service lifecycle steps and some eTOM process groupings.

Service lifecycle step	eTOM process	Comments
Service Requirement Analysis	Product Lifecycle Management (Service Development & Management)	
Service Specification	Product Lifecycle Management (Service Development & Management)	
Service Design	Product Lifecycle Management (Service Development & Management)	*This step is detailed below.*
Negotiation & Agreement with suppliers	Supply Chain Development & Management	Registration of SLA and interfaces.
Service Implementation	Product Lifecycle Management (Service Development & Management)	Service ready to be deployed in a specific environment.
Test (before deployment)	Product Lifecycle Management (Service Development & Management)	

Service Instantiation	Operations Support & Readiness (Service Management & Operations)	Service Instantiation is the first step of the deployment process in order to make the service ready for user subscription and then to be activated for usage.
Deployment Test	Operations Support & Readiness (Service Management & Operations)	
Service Subscription	Fulfilment	
Service Activation	Fulfilment	
Service Usage		This is not part of eTOM and is part of Service Delivery
Data Collection Analysis & Control	Resource Data Collection & Processing	
Service Quality Management	Assurance (CRM, Service Management & Operations, Resource Management & Operations, Supplier/Partner Relationship Management)	
Reporting	Customer Interface Management, Billing & Collections Management	

Table 1: Service Lifecycle steps mapping to eTOM Business Processes

In the rest of the section we focus on the service design step as it is the place where the models will be defined. It is therefore the step which highlights our modelling approach.

Service Design Process and Service Models
In the figure below we summarise external interactions to the Service Design Process.

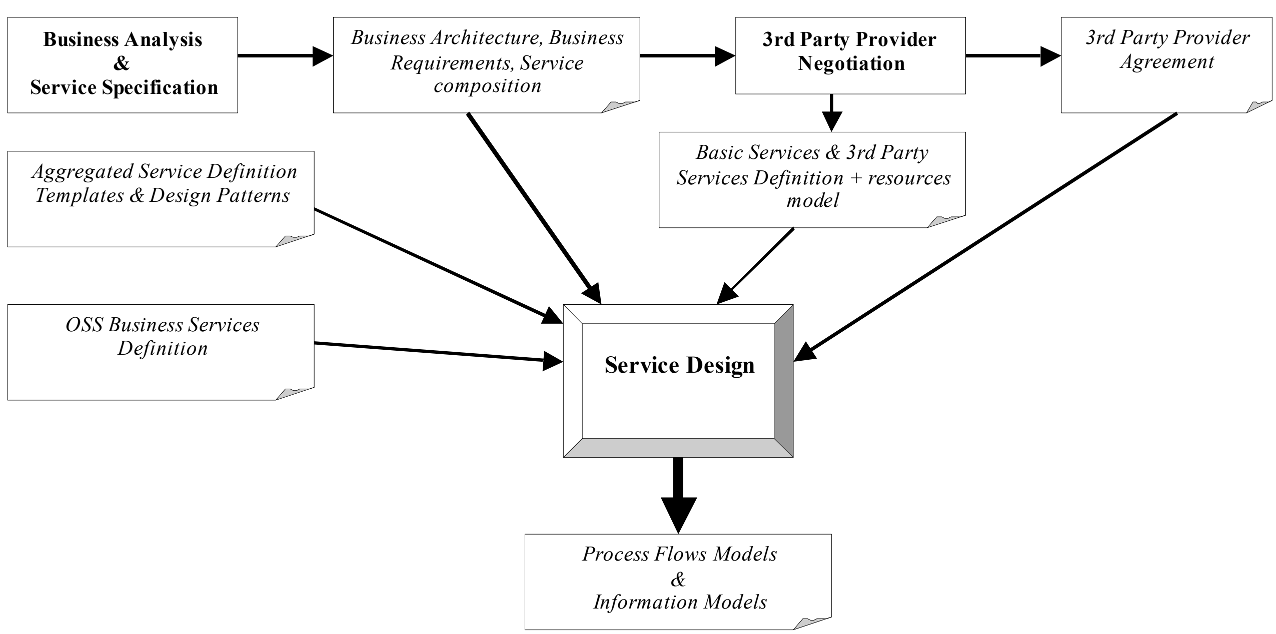

Figure 3: Service Design Process: External Interactions

The Service Design process depends on outputs from previous lifecycle steps. The starting point comes from the *business requirements*. The *business analysis* defines the *business architecture* capable of supporting the delivery of the service. It includes therefore a definition of the relationships with 3rd party providers, in terms of requirements.
This first analysis result needs to be confronted with the state of affairs, i.e. existing "internal" services and potential 3rd party services. This will enable *service composition* to be defined, i.e. what are the service elements comprising the aggregated service. This result is obtained from iterations.
From this step, 3rd Party Providers have been defined and it is therefore necessary to negotiate with these 3rd Party Providers to define *3rd Party Provider Agreements*. This step can be desynchronised from the previous one. Thus the Service Aggregator can build new aggregated services including service elements from 3rd Party Providers, based on an existing agreement. The *3rd Party Provider Negotiation* process could be mapped to ebXML processes and the 3rd Party Provider Agreement could be mapped to CPA (Collaborative Protocol Agreement).

The main challenge in the service design process is to map the need (the business requirement) to the existing and reusable components. This goal can only be reached through an iterative approach.

The main concepts we manipulate here are:

- *Services and service components*, defined as entities that can be provided to the customer/user. A set of Business Process flows based on DBC (Deployable Business Components) is integrated.
- *Business Process Flow* is, in general, an atomic interaction with the customer/user and defines the business logic between DBCs to support such interaction.
- *Deployable Business Components*, reusable entities that encapsulate a defined set of functions related to business and its associated data. Each DBC defines a reference point (a set of contracts) for integration that defines how to access its functions and data.

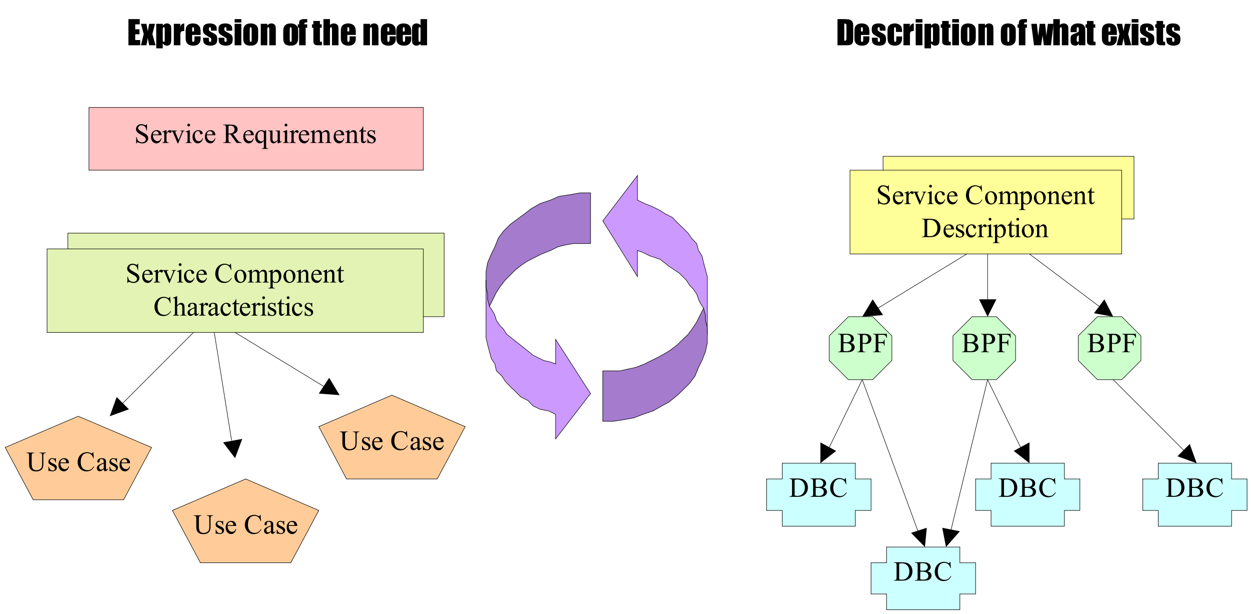

Figure 4: Design Process, mapping from the need to what exists

As presented in the figure above, the need is defined from the service requirements, which can be decomposed logically into service component characteristics. These are described from a suite of Use Cases, describing the behaviour of the service component in different situations.

Below is an example of service decomposition into high-level service components.

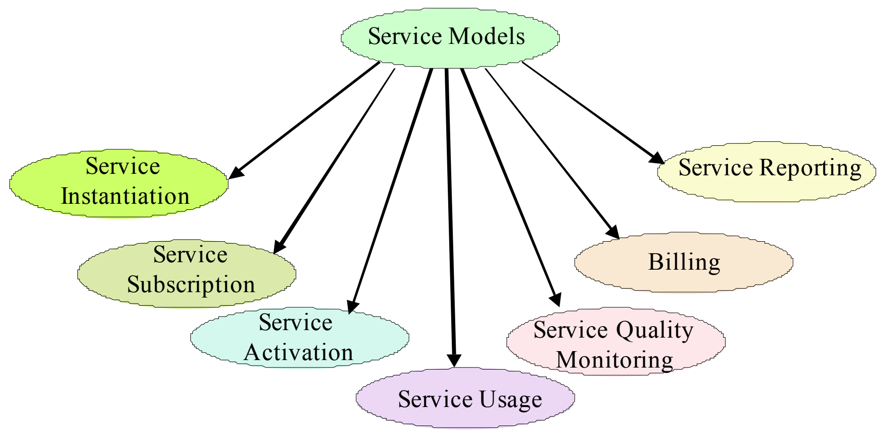

Figure 5: Service decomposed into Service Components

On the other hand, i.e. what already exists and will be used to build the service, there is a set of existing service components. Based on the description of these entities, we try to check if some of them correspond to the service component characteristics as described in the need. If this is the case, it means we can take the whole existing service component as it is to build this part of the aggregated service. Otherwise, we need to go to a deeper level, i.e. looking at BPF and DBC to build the service component. In this case we need to define BPFs from the Use Cases. This allows the mapping from the business level to the system level. For the BPF specification, in the DIS sub-trial, we used high-level sequence diagrams (see Figure 6), i.e. interactions between DBCs. This exercise is done iteratively to map the need from what exists, i.e. the DBCs and their contracts.

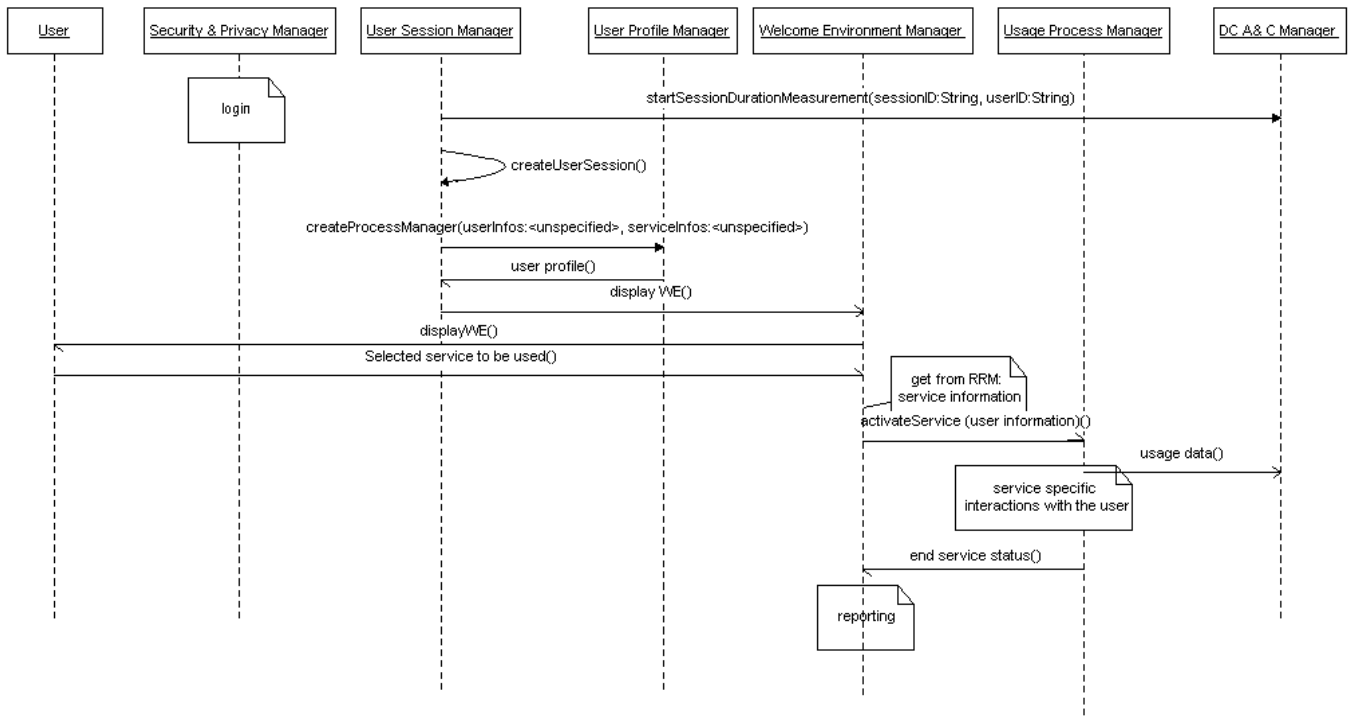

Figure 6: Example of high-level sequence diagram (part of the service usage)

The Business Logic could have been described with a UML state diagram as well.

To resume, at this step we get:

- The breakdown of the service into service components, based on the definition of use cases.
- Some of the service components already exist, and thus they can be reused as they are.
- Other service components need to be specified, this is done by defining the business logic, based on sequence or state diagrams which define the interactions between DBCs.

The next step in the service design process concerns the definition of the information to be shared between components. In the AlbatrOSS project we adapted the SID model defined by NGOSS and extended it with Presence and Availability management, Service Aggregation and user preferences definitions. The use of SID models in AlbatrOSS is detailed in [Alba_NGOSS].

At the end of the service design process, we obtain the models allowing the service to be implemented. The *service implementation step* is part of the service development phase and consists of the mapping of models into specific formats.
For the DIS sub-trial, we mapped sequence diagrams describing the business logic into Petri networks (see Figure 7). These have then been injected into the Process Manager during the deployment phase. We could have used here [BPEL4WS], which is today recognised as a good candidate to implement process flow between Web Services. We did not make this choice as we were able to reuse an existing Process Manager developed by Telefonica I+D.

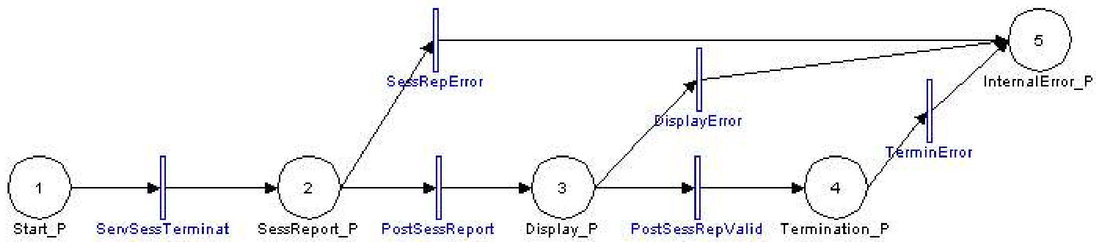

Figure 7: Example of Petri Network for Business Logic implementation

During the service implementation phase, we implemented the service, with the definition of KPIs (Key Performance Indicators) and KQIs (Key Quality Indicators) with HP SQM, allowing SLA and performance to be managed.
An interesting point is that even the deployment phase can be defined through Business Process Flow and therefore can be partly automated.

This section presented an approach towards building services from the definition of the service lifecycle and the modelling of the service during the design phase.
The next section presents the architectural aspects of the platform where services will be run.

Service Delivery and OSS Architecture

This part is looking at the definition of the architecture of a platform enabling service delivery and operations support. For this we integrated some of the requirements defined by the TM Forum for Next Generation OSS. This can be summarised by two main points:

- The possibility to plug new components into the system (without modifying components already plugged in).
- Automate the different processes as much as possible.

We added new requirements to the definition of the platform:

- Integration of new functions (not defined in the eTOM Business Process Framework) in support of service delivery. Therefore, the platform is defined as a Service Delivery and Management platform.
- We used the concept of aggregated service as one service delivered to the end user resulting from the aggregation of different service components, which can be provided by third party providers. We need therefore to support interactions with suppliers through the platform.

The main NGOSS principles reused in the definition of the platform were:

- We define a component-based approach, which implies that each component provides a contract and that components are loosely coupled.
- The components are defined as business components and therefore interactions between components can be defined as business logic flows. Moreover, the information exchanged and shared between components is based on common models: the SID models.

The main principles of the Service Delivery and Management platform are:

- All components interact through a common bus, which allows adding, removing or replacing components with minimal impact on the other components.
- We categorised components into 3 different levels:
  1. ***The framework services***, grouping a set of functional blocks providing facilities for the integration and operation of the components.

2. ***The business components***, which are dedicated to specific business processes as described by eTOM (Ordering, Subscription, Billing, Service Quality Assurance, …). They support delivery and operation of the services.
3. The top layer corresponds to ***the core service components*** and includes service elements. Thus, they provide the service content.

The overall view of the Service Delivery and Management architecture is presented below.

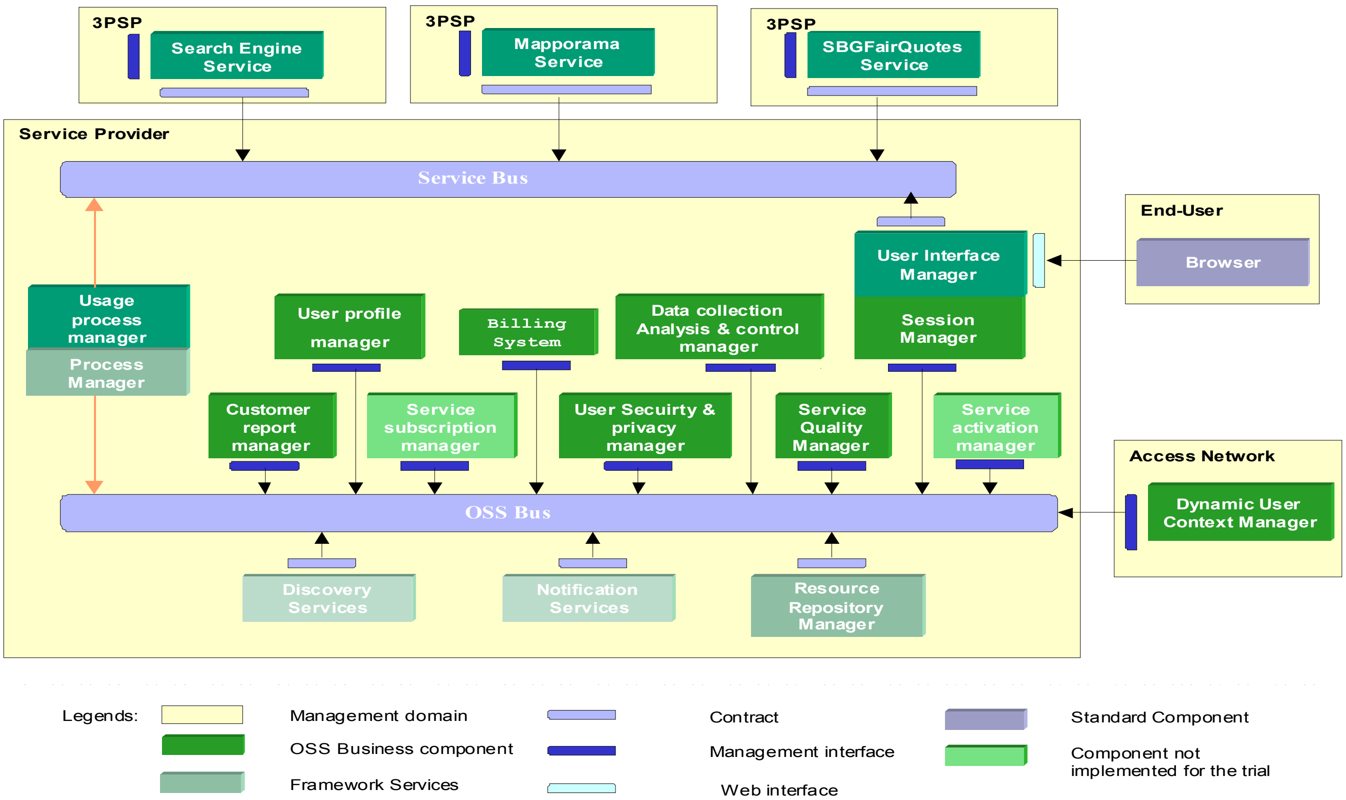

Figure 8: Components of the Service Delivery and Management platform

Based on this architecture we can show that:

- "business components" can be added or removed, or even replaced, quite rapidly,
- new services can be deployed, activated, used and operated effectively and a lot of the tasks can be automated.

We present now a quick tour of the main components of the architecture (a detailed presentation of the components and the overall system can be found in the AlbatrOSS deliverables available from the AlbatrOSS *web site).*

The "framework services" are defined at the lowest level, they include:

- the *process manager*, which executes the business logic as defined for each service,
- the *resource repository manager*, allows business components to share common information such as SLA,
- the *notification services*, implement notification mechanisms allowing business components to communicate via events,
- the *discovery services*, provide yellow and white page services to access component references.

Some other framework services could have been defined, such as services enabling the registering of and secured access from third party providers.
All framework services provide Web service interfaces.

The platform implemented for the DIS sub-trial integrates a large range of business components. Some of them have been implemented from scratch, others were already implemented (the billing system from WIT) and we also integrated commercial products such as HP OV SQM. This allowed us to show that, as long as the interfaces are provided, the business logic can then be defined during the design phase, and based on this you inject this business logic into the process manager to deploy the new service.
Therefore, we are not in a one-step process where you design the service, integrate components and deploy the service from this specific configuration. Here we plug in the components (independently of each other) quite separately from the design and deployment of the service. The service is more oriented towards a business logic definition, which integrates an abstract view of the components. There is, therefore, a loose coupling between the components and the service, allowing rapid service creation (However, realistically some specific integration of the service needs to be done with some of the components).
This conceptual view is fully supported by the definition of Shared Information Models as done by TM Forum with SID. We adapted some of the SID models for the DIS sub-trial.
We should also consider the definition of standardised interfaces, at the business level, allowing the business logic to be described independently of the specific components integrated into the system. The OSS/J initiative, which is in the process of integration with TM Forum work, provides a first implementation in the form of APIs. This could be reused as a starting point.

One important part of the architecture, which comes from a business request, concerns the integration of third party providers. These provide content, application, connectivity or network services. They also provide Web services interfaces. Thus, they can be easily integrated, but WSDL is not sufficient to enable B2B. Different ongoing work still need to be integrated. First, the relationships defined here are based on a B2B business model, therefore OASIS work around ebXML could be reused here, particularly for the definition of the Collaboration Protocol Agreement. Each of the third parties integrated into the DIS sub-trial provide core service interfaces plus a service management interface, allowing the Service Aggregator to ensure end-to-end management of the service (for the fulfilment, assurance and billing processes). IBM aims to integrate the SLA into the Web services definition, this is done in the Web Service Level Agreement project [WSLA]. DAML-S [DAML-S] work should also be considered in the future for the definition of Web Services in a semantic Web, this should allow service content to be qualified and therefore to enable on-the-fly service aggregation or service discovery.
There is therefore lot of work that should be done in the future to implement fully the service aggregation concept.

The last point concerns the connectivity services as well as network services. There are a number of groups working on the subject today. The objective is to build, deliver and operate services independently of the underlying networks. This would allow rapid service creation as well as, for the user, the possibility of accessing services independently of the access network. There are different and more or less complementary possibilities of implementing this. In the DIS sub-trial we encapsulated the connectivity services through the session manager component. Connectivity has been restricted to IP and we used the Instant Messaging Service to send an SMS when the user is not reachable. On the other hand, we retrieve user information provided by the network. This is done through Lucent's OSA/Parlay Gateway (MiLife ISG). We recognise that a big part of the work here still needs to be done, but lots of working groups and projects are already investigating this area.

Conclusion

The assessment of this work has been done through the implementation of a concrete scenario. You can have access to a live demo from the AlbatrOSS web site.

The approach we have considered in line with TM Forum and SOA (Service Oriented Architecture) is to bring the service to the forefront. This means the relationship between the user/customer and the service provider is the driver, and the service can be adapted from the definition of this relationship. To support this we propose a new way of creating services, integrated into the service lifecycle phases. We also defined the architecture of a Service Delivery and Management platform, which repositioned at the same time the service platform and OSS.

Obviously this work is a starting point, however, we think all the technical ingredients to support such an approach are there. In the DIS sub-trial of the AlbatrOSS project, we undertook a first validation. Moreover, aspects related to NGOSS architecture principles have also been validated by some catalyst projects of the TM Forum.

References

[Parlay_Group] Parlay 4.1 specifications: http://www.parlay.org/specs/index.asp

[OSS/J] OSS/J Initiative: http://java.sun.com/products/oss/

[ParlayX] Parlay X working group: http://www.argreenhouse.com/parlayx/

[WSLA] IBM – Web Service Level Agreements project: http://www.research.ibm.com/wsla/

[ebXML] OASIS group: Enabling a Global electronic Market: http://www.ebxml.org/

[DAML-S] http://www.daml.org/services/

[eTOM] enhanced Telecommunications Operation Map: http://www.tmforum.org/

[Alba_NGOSS] White Paper: NGOSS Principles used in AlbatrOSS: http://www.ist-albatrOSS.org

[BPEL4WS] http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/