Nikolaos Kosmas and Kenneth J. Turner. Requirements for Service Creation Environments. In Ignac Lovrek, editor, Proc. 2nd International Workshop on Applied Formal Methods in System Design, pages 133-137, Zagreb, June 1997.

Requirements for Service Creation Environments

Nikolaos Kosmas and Kenneth J. Turner
Department of Computing Science
University of Stirling
Stirling, Great Britain
{nko,kjt}@cs.stir.ac.uk

Abstract

Service Creation Environments are the new frontiers in telecommunications. Efficient and reliable service creation is vital towards the evolution of the telecommunication domain as we move into Intelligent Networks. This paper discusses the concept of service creation environments and how it is related to the process of service creation. Additionally, a list of abstract requirements for service creation environments is outlined.

1 Introduction

Service Creation Environments are the new frontiers in telecommunications ([7]). The notion of service creation environments is strongly connected with the concept of service intelligence in telecommunication networks. Introducing intelligent networks, networks that enable the user to customise and manage "calls", had a double effect in enterprise terms: on the one hand technological advances made call costs almost negible and satisfied customer demands for performance [1], but on the other hand competition forced providers to offer more service choices in order to come up with the lost revenue. The need for fast development of (telecommunication) services created two new concepts within the field: Service Vendors who specialise in the creation of services, and Service Creation Environments which represent the platforms that enable "mass" production of services. In this perspective the success of either Intelligent Networks or Software Creation Environments heavily depends upon the success of the other [7].

2 Service Creation and Service Creation Environments

One of the most researched issues within the context of (tele)communication networks is the notion of service life-cycle. Services are the corner stones of Intelligent Networks and therefore a subject of crucial importance not only to Service Providers (SP) and Service Vendors (SV) but also to all the other interacting parties. A number of models of the service life-cycle have been formed, exhibiting little differences. Figure 1 provides one of the more abstract overall views of service life-cycle as presented in [1].

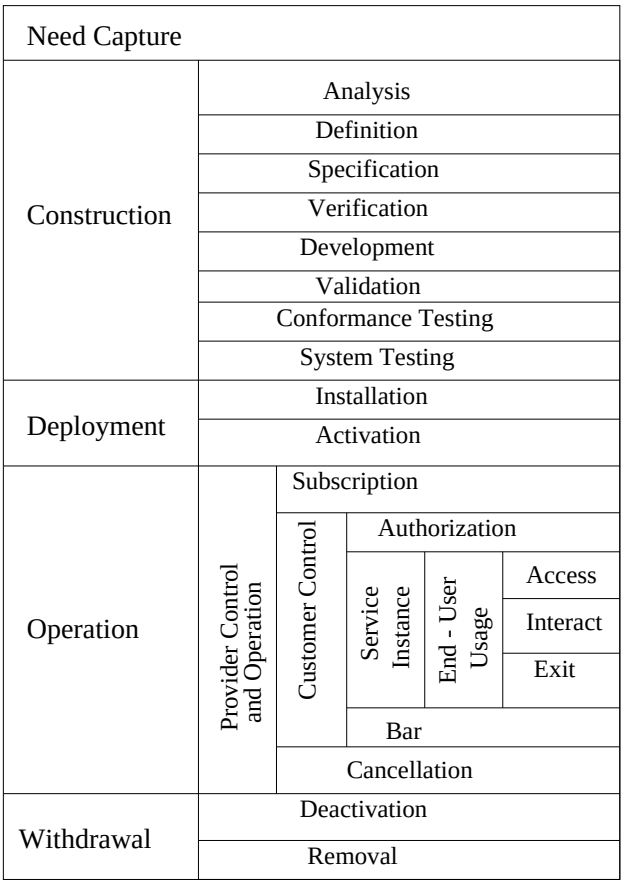

Figure 1: Service Life-Cycle

Taking another perspective we can group certain of the phases shown above and create more general phases of the service life-cycle that relate to activities

[1] Examples include connection times or reliability of voice/data transfer.

found in other domains [2]. Grasdijk and Mudhar ([6] and [8]) identified three main stages: the service creation phase, the service provisioning phase and the service usage phase [3]. Within this viewpoint, service creation includes all the activities necessary in order to create a new service type, not the physical service instances, either from existing services and service components or starting anew.

The logical conclusion of the above observations is that the service creation activities [4] fall into the scope of the service creation environment. Before we go on and accept this statement we consider it useful to try and define, more accurately describe, the concept of a service creation environment. In our view a service creation environment can be seen as the logical and physical framework where the building of services takes place. Logical because it includes the methods and techniques used during service construction and physical because it takes place within specific organisations or parts of organisations [5]. With that in mind the next issue to be addressed refers to the limitations of the scope of service creation environments. Without being assertive we can assume that there are certain activities directly related to service creation that cannot be effectively performed within a service creation environment due to the limited focus that it is available in terms of the number and type of services associated with each service creation environment. For example if we consider the Feature Interaction Problem ([4], [3]) then we realise that verification of service behaviour is incomplete no matter what techniques or tools are used.

2.1 Service Creation Life-Cycle

It is not our intention to reproduce models of the service life-cycle in this section. We merely concentrate on the most important activities and present the role they hold within the context of service creation.

Requirements capture and analysis is the obvious starting point. The set of requirements that results from the marketing research, conducted by the service provider, on the business feasibility of the service is analysed and omissions or ambiguities are identified. Specification of the service will complete the analysis process and lead to the derivation of the first service prototype. The stage of specification is crucial not only with respect to service creation in itself (engineering–wise) but also because it affects other phases of the service life-cycle. Leaving out for the moment verification of services within an environment that contains other services as well, we emphasise the need for reuse in all service creation stages. As for software engineering, specification reuse can prove to be vital towards limiting both the time and cost needed for the development of the service. Also during specification we have the last link of communication between the service vendor and the service provider before the product is tested and validated in the execution platform.

The subsequent stages in service creation involve the realisation of the service within the context of a specific execution platform (information that should be produced during analysis of the service). What is interesting here is that current service creation environments have a restricted scope. This is why they are called dedicated service creation environments, both in terms of the services created (for example Capability Set 1 of the IN) and the platforms that are meant for execution (general-purpose computers or specialised switching systems). Nevertheless, the need for generic service creation environments has increased critically if we want to reach the goal of an open telecommunications market.

2.2 Structure and Interfaces of Service Creation Environments

A service creation environment is basically a platform where services are built, ideally using a library of components and established "plug–in" methods. Since, in the abstract, it is an environment within an organisation, it is influenced by the culture and general philosophy of the organisation. It is also subject to the limitations of the organisation in terms of tool and personnel support.

Building a service creation environment is an incremental process that depends on the nature and number of services that have to be developed (unless we create a library of service components without having specific services in mind!). The process itself bears some resemblance to the spiral model of traditional software engineering. However, it is not only the infrastructure that gets enriched (service components) at each step but also the tools, the processes and the methodologies (service analysis and verification). Ponten ([10]) has described such a process of creating a SCE. The service creation environment domain consists of the services already created, the technology used and the organisation culture. Each step in the process results in the development of a more advanced SCE which can be used, at least in theory, for the development of services from different domains. Progressively, the service creation environment is enriched and the effectiveness is improved. Note that though there seems to be no profound bar-

[2] For example imagine services as cars built by a car manufacturer.

[3] Mudhar uses the terms deployment and utilisation instead of provisioning and usage.

[4] By activity Mudhar implies "a procedure that uses a set of concepts, methods and solutions".

[5] The case where a SP develops its own services.

rier in the evolution of the environment, the original limitations imposed by the organisation still apply.

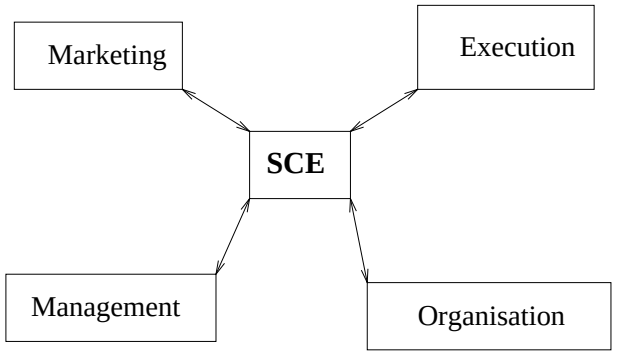

Figure 2: Interfaces to Service Creation

Within the same work Ponten also identified interfaces to service creation shown in Figure 2. The interaction depicted by these interfaces complete the notion of service creation in conjunction with the pure engineering process of achieving the right behaviour of a service. What should be also interesting is to examine the interfaces of the organisation itself, represented by the service vendor, with the other enterprise entities of the telecommunication domain. The main interaction that occurs is between the service vendor and the service provider. The vendor is contracted to build service(s) for the provider fulfilling specific requirements. At stages during the creation process the vendor needs information both on the network platform on which the service is going to operate and a user view of the service behaviour. The dashed lines in Figure 3 indicate that this information probably comes via the contractor and not the source parties. Furthermore, the vendor might develop the requested services in line with international standards. Overall control over the above interactions is kept with the national regulatory bodies.

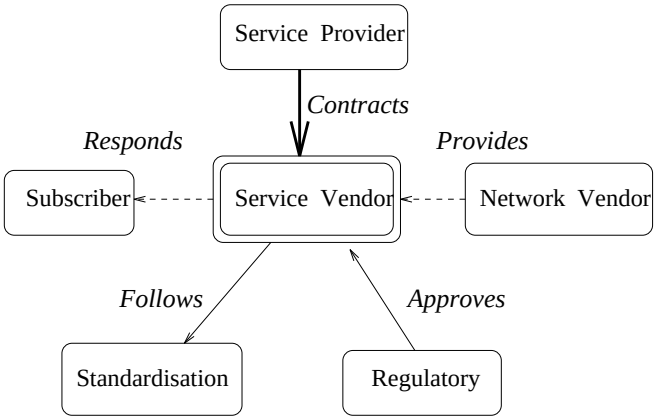

Figure 3: Interfaces of the Service Vendor

3 Abstract Requirements for Service Creation Environments

We have outlined so far the aspects of service creation environments and how they are related to the service life-cycle. More useful results can come from pilot implementations of service creation environments like the one developed within the EURESCOM project P103 "Evolution of the Intelligent Network". What is left in order to provide an overall view of service creation environments consists of the expectations that, mainly, service providers have and the requirements, in an abstract form, that are considered critical for their [SCEs] successful application.

3.1 Expectations

Let us summarise the main current demands in telecommunications. We can identify two main axes of action: the need to introduce services rapidly and efficiently, and the need to guarantee that the operation of these services will not produce unwanted side-effects within a distributed telecommunications context. How these are translated into goals with respect to the service life-cycle mentioned in earlier sections is quite straightforward. Assuming that services are developed without any subsequent execution side–effects, we can focus on the service creation phase. This is so because, ideally, adding a service to an intelligent network is relatively easy, making use of the concept of centralised service control within the Service Control Point (ITU Recommendation Q.1205). Access to the service and management of service operation does not fall, not directly, under the scope of service creation environments.

What is expected from an efficient service creation environment is the capability to build services rapidly even if marketing considerations result in changes of requirements during service construction. Services must be developed not on a stand alone basis but as classes of services, and in such a way that further modifications can be incorporated smoothly in the previous versions without having to change structural elements. In addition service creation should be decoupled from implementation details that concern the execution platform as much as possible providing the opportunity to reuse part of the process and results of service construction in order to make the service available for execution in different networks. In addition, the influence of the organisation culture must be such that it would not prevent the contractor (service provider) from enhancing the service within the context of another service creation environment.

What we excluded so far was potential inconsistencies between the blueprint behaviour of the service

and the actual behaviour as observed by the user of the service. It is to be considered that service execution within a "model" environment is in some cases totally different from the one produced within a real environment which includes competition for resources or conflicting interests. These new considerations open another dimension in service verification and system testing that is in the centre of service creation.

3.2 Meeting the Objectives

What service creation environments can offer towards the satisfaction of the above needs lies with the efficient development of classes of services. A single service satisfies only one class of customers, leaving therefore a "service gap". In the highly competitive telecommunications market this is an unacceptable approach. Instead marketing research concentrates on recognising core services that can be differentiated using optional service features, thus creating classes of services that exhibit similar behaviour by inheriting some core functionality. Even if there is not a direct correspondence, a class of services satisfying a particular need (for example answering all calls) can have a dramatic improvement on the success of the service at relatively low cost and time delay. Enterprise related considerations concerning the identification, analysis and specification [6] of services and service requirements is a task for the service provider. Within the service creation environment effort should be directed towards processing the service requirements in a more formal and systematic way in order to ensure at the earliest possible stage the consistency and completeness of the requirements.

Assuming that the service creation environment has evolved to a point where the construction of service behaviour is an automated procedure the main task that the service creation engineer faces is the one of validation and verification of the produced behaviour against the required behaviour and the user's perception of the service behaviour. It would seem more proper to move this task over to the service provider, who has a general picture both on the services already deployed and the characteristics of the execution platform However, we note two main critical points. The first is that the service vendor having an overall picture of service development can pinpoint and assert any misbehaviour more accurately than the receiving service provider and second that the question of service interaction can be, in some cases, translated into one of interaction of low-level service components leading to a better verification strategy. And keep in mind that the service vendor is the one that would have to re–develop the service in case the behaviour is not the one originally intended.

How does a service vendor meet the above goals? We merely provide a set of abstract requirements which, if met, we believe can prove critical towards structuring service creation environments that can fulfil the aforementioned objectives:

- reuse is essential, a library of components must be established at all levels. Abstract specification components are prefered in comparison to more concrete implementation ones. This procedure (creating services by plugging-in components from a library and enriching the library of components with each new class of services) should apply to both specification and realisation phases.

- reuse and construction principles specific to service architectural characteristics have to be established. In this way the philosophy of the service architecture [7] can be incorporated in the service creation environment.

- a high degree of automation is needed. Languages for specification and code generation that are supported by reliable tool sets are necessary for the verification both of the service specification and the conformance of the implementation. Formal Description Techniques [8] have been used successfully within a telecommunications context ([5], [9] and [2]) and appear as a logical choice.

- an interface to the marketing developments is crucial for reducing the time-development factor. Furthermore, a uniform medium of communication between the service vendor and the other interacting parties has to be developed. Formal methods provide the necessary degree of precision and tractability that ensures better exchange of information. Bear also in mind that generic service and network constructs within the ITU Recommendations have been specified using SDL diagrams.

- feedback from network providers and service users is crucial. The library of components and combinators should not be dependent to specific switching systems and the service behaviour has to be validated from a user perspective.

- traditional software engineering practices should be enriched by new ones (service engineering, distributed processing). Suitable conceptual models of services and service contexts result in flexible analysis and design components. The

[6] Informally using natural languages.

[7] An example of a service oriented architecture in the IN.

[8] LOTOS, SDL and ESTELLE.

overall development methodology should support the conceptual framework.

- experienced personnel that can communicate with the service provider in the same "language" are critical towards creating services that correspond to the original specifications.

- verification procedures have to be uniform starting with the first group of services to be developed. This does not mean that these procedures will not be explored to add new possibilities. Focus should be directed towards developing mechanisms for the detection and resolution of unwanted service interactions.

In our opinion a service creation environment is not only the physical area where a service is developed but also is the context for the exploration of the concept of service itself.

4 Conclusions

We have seen the requirements of service creation environments and the role that these frameworks have with respect to modern telecommunication needs. It is clear that SCEs have been made necessary both by the diversity of telecommunication needs and the current demand of new classes of services. Effective service creation environments can offer a competitive edge in the specification and analysis of services. The move from today's dedicated environments to more open and generic service creation environments is necessary if we want to keep intact the pace of evolution towards and beyond Intelligent Networks. Reuse and component service development are the key issues for the success of SCEs.

The approach of moving the service creation responsibility to service vendors, as part of the development of an open telecommunications market, can be greatly enhanced by the use of formal methods for the specification of services. Results from the several service creation phases can be communicated effectively among the interested parties. The process of analysing potential undesired service interactions can be enhanced enormously using the mathematical power of formal specification and the capabilities that the underlying tool sets offer for simulation and prototyping.

References

[1] Hendrik Berndt, Peter Graubmann, and Masaki Wakano. Service Specification Concepts in TINA-C. In Hans-Jurgen Kugler, Al Mullery, and Norbert Niebert, editors, *Towards a Pan-European Telecommunications Infrastructure – IS&N '94*, pages 355–366. Springer-Verlag, 1994.

[2] J. Bredereke and R. Gotzhein. A case study on specification, detection and resolution of IN feature interactions with ESTELLE. Technical Report 254, University of Kaiserslautern, May 1994.

[3] Jane Cameron et al. A Feature-Interaction Benchmark for IN and Beyond. *IEEE Communications Magazine*, 31(3):64–69, March 1993.

[4] T F Bowen et al. The Feature Interaction Problem in Telecommunication Systems. In *Seventh IEE International Conference for Software Engineering in Telecommunication Systems*, pages 307–319, July 1989.

[5] M. Faci and L. Logrippo. Specifying Features and Analysing their Interactions in a LOTOS Environment. In L. G. Bouma and H. Velthuijsen, editors, *Feature Interactions in Telecommunications Systems*, pages 136–151. IOS Press, 1994.

[6] Marjan Grasdijk, Jan Dreteler, Harold Braux, and Jean-Louis Le Bail. Modelling Services in the Portfolio from a Service Provisioning Perspective. In Hans-Jurgen Kugler, Al Mullery, and Norbert Niebert, editors, *Towards a Pan-European Telecommunications Infrastructure – IS&N '94*, pages 133–143. Springer-Verlag, 1994.

[7] E Fletcher Haselton. Service Creation Environments for Intelligent Networks. *IEEE Communications Magazine*, 30(2):78–81, February 1992.

[8] Parminder Mudhar. A Service Creation Environment for a future Intelligent Network. In Al Mullery Hans-Jurgen Kugler and Norbert Niebert, editors, *Towards a Pan-European Telecommunications Infrastructure – IS&N '94*, pages 333–342. Springer-Verlag, 1994.

[9] A. Nyeng and B. Moler-Pedersen. Approaches to the specification of Intelligent Network Services in SDL-92. In *Proceedings of the Sixth SDL Forum*, pages 427–440. North Holland, 1993.

[10] Lars Ponten, Joacim Hallstrand, and Maria Manuela Marques. Building Dedicated Service Creation Environments for Reuse based Production. In Hans-Jurgen Kugler, Al Mullery, and Norbert Niebert, editors, *Towards a Pan-European Telecommunications Infrastructure – IS&N '94*, pages 169–178. Springer-Verlag, 1994.