SERVICE CREATION TECHNOLOGIES IN OPEN PROGRAMMABLE NETWORKS

Dionisis Adamopoulos
Department of Technology Education & Digital Systems, University of Piraeus, Greece

Constantine Papandreou
Hellenic Telecommunications Organisation (OTE), Greece

Keywords: Service creation, programmable networks, new telecommunications services, service engineering

Abstract: The architectural frameworks that support the development of advanced telecommunications services have a generic and abstract character and do not specify with accuracy the proposed activities and the order of the steps that are necessary for their successful realisation. For this reason, this paper proposes an incremental and iterative service creation methodology, argues about its usefulness, outlines its basic characteristics and focuses on the necessity to complement it with a suitable Service Creation Environment (SCE). Therefore, the paper attempts to define SCEs, identifies their main characteristics, examines important related approaches in a critical manner and reasons about the relation of the SCE with the proposed service development methodology, highlighting the role and purpose that it should have.

1 INTRODUCTION

An architectural framework for telecommunications services is by its definition an abstract entity, which consists of a set of concepts / principles and a set of guidelines and rules. For this reason, it is usually more descriptive rather than prescriptive, and its application can be a complex task. Furthermore, there seems to be no end to the emergence of new services, each requiring a new set of communications capabilities. In a world already replete with a multitude of services, the addition of new intricate services can be a daunting challenge.

For this reason, this paper argues that current service creation practices are not able to fully satisfy the need of service development in an environment open to competition and open to changes in market and technology and proposes a specialised incremental and iterative service creation methodology (Adamopoulos, 2002). After examining briefly the main phases of the proposed methodology, the paper focuses on Service Creation Environments (SCEs), attempting to define them and identify their main characteristics. In order to reason about the role and purpose of the SCE in the proposed telecommunications service development process, the paper examines important related approaches in a critical manner, highlights the relation of the SCE with the service development methodology, and explains under which circumstances the SCE is able to automate efficiently the service creation process, without semantic loss, with the use of appropriate, carefully designed and tested, customisable, and user-friendly software tools.

2 THE NEED FOR METHODOLOGICAL SUPPORT

Considerable effort is being devoted in Europe and world-wide to the definition of advanced service creation practices. The necessity for appropriate methodological support during service development is highlighted by both OSA (Open Services Architectural framework), which clearly states that "… only the combination of architecture and methodology will meet the required expectations on long-term efficiency of service creation, provision and management", and TINA-C (Telecommunications Information Networking Architecture – Consortium), which emphasises its importance. Furthermore, both these two architectural frameworks recognise the

Adamopoulos D. and Papandreou C. (2004).
SERVICE CREATION TECHNOLOGIES IN OPEN PROGRAMMABLE NETWORKS.
In *Proceedings of the First International Conference on E-Business and Telecommunication Networks*, pages 294-299
DOI: 10.5220/0001394002940299

need to (gradually) leave behind service creation practices that are market and technology dependent, as it is clear that they can't address the requirements imposed on service engineering activities by the emerging telecommunications environment.

The most important and the most detailed from all the approaches is the one proposed by the ACTS project SCREEN, which focuses on component-based service creation (ACTS, 1997)(ACTS, 1998). More specifically, according to SCREEN, the construction of a service should ideally consist of the simple activity of assembling together already built "pieces" of services commonly referred to as components, which are the units of construction and reuse. For this reason, SCREEN has established a component library that enables the storage, management, and easy browsing and retrieval of reusable component representations. For building new components and for customising the already existing ones, SCREEN provides a (kind of) service creation methodology. However, this methodology (which is actually a service creation practice), except from being abstract, focuses mainly on the service design phase, underestimating all the previous phases, and is heavily based on SDL, ignoring UML. Therefore, the SCREEN methodology can be considered as a loosely-coupled framework, incorporating mostly service design related guidelines and technologies that fail to efficiently support the entire service creation process.

Additionally, important disadvantages of the SCREEN approach, that characterise in general component-based service creation practices, are the following:

- Service developers focus on the service design underestimating or even completely ignoring activities related to the elicitation of requirements and service analysis. In this way, the possibility of having a service design that doesn't reflect (correctly or even at all) the real service requirements, and thus a telematic service that is not acceptable by its users, is increased.
- Maintaining a (not trivial) telematic service that has been constructed by assembling reusable components is a difficult task, because the semantic origin of these components is not known. Due to the lack of (proper) requirements elicitation and service analysis results, when a feature of the service has to be changed, the components that should be modified can be found only in ad-hoc ways.
- Criteria for determining the usefulness and suitability of a reusable component when creating a telematic service are not specified. Therefore, the possibility for poor choices that can lead to unnecessary complex service designs (especially when the service developer is not experienced) is increased.
- There is significant dependence on the quality and completeness of the available library of reusable components. In general, it is very hard to foresee in advance all the different needs of services to be constructed and have the right (properly designed) reusable components available. Furthermore, there is a lack of commercially available service component libraries and the efficient management of these libraries is still a significant unsolved problem.
- Creating carefully designed reusable service components is a difficult task and can increase significantly the cost for creating a telematic service. Thus, there is always a temptation to develop a service as quickly as possible even at the expense of future reusability.

From these remarks, and taking into account the inadequacy of general purpose Object-Oriented Analysis and Design (OOAD) methodologies for service creation purposes, the necessity for a service development methodology is evident. Such a methodology can be combined with component-based service creation practices, offering an efficient, realistic, integrated, and systematic approach for the development of telematic services. Finally, the need for a service development methodology is reinforced when considering ODP, which offers a conceptual background (a meta-framework) by providing a reference model and powerful concepts to support the specification and design of distributed systems. However, ODP does not provide any methodological support to facilitate the modelling inside each viewpoint or to enable the specifications in different viewpoints to be linked. It does not offer a methodology that starts from enterprise views and proceeds through all the viewpoints establishing the appropriate models in an integrated manner.

In order to meet the challenges identified in this section, a service creation methodology is proposed with the intention to accelerate the service life-cycle so that new and enhanced services can be developed and deployed at a faster rate, in a cost effective manner, without making quality compromises in an open deregulated multi-provider telecommunications market place. This methodology, given a set of requirements that a service should meet, a set of the available service independent features (normally in the form of service components), and a target DPE wherein the service will be deployed, facilitates the design and implementation of a telecommunications service, which meets the desired requirements by promoting the use of the service independent features (Polydorou, 1998).

3 OVERVIEW OF THE PROPOSED SERVICE CREATION METHODOLOGY

Telecommunications operators need to master the complexity of service software, because of the highly diversified market demands, and consequently, because of the necessity to quickly and economically develop and introduce a broad range of new services. To achieve such an ambitious, yet strategic to the telecommunications operator's goal, a service creation methodology based on the rich conceptual model of TINA-C is proposed.

The proposed service development process is based on an iterative and incremental, use case driven approach. An iterative service creation life cycle is adopted, which is based on successive enlargement and refinement of a telematic service through multiple service development cycles within each one the telematic service grows as it is enriched with new functions. More specifically, after the requirements capture and analysis phase, service development proceeds in a service formation phase, through a series of service development cycles. Each cycle tackles a relatively small set of service requirements, proceeding through the phases of service analysis, service design, service implementation, and service validation and testing. The telematic service grows incrementally as each cycle is completed.

In the following paragraphs, the use of a SCE together with the service development methodology is proposed, in order to realise its full potential and assist the service developer(s) when applying the methodology by automating and simplifying as much as possible the service creation process, and by facilitating consistency and verification checks.

4 SERVICE CREATION ENVIRONMENTS

The importance of SCEs, combined with an appropriate service development methodology, is constantly increasing because, in the deregulated and highly competitive telecommunications market, the success of service providers is determined by the efficiency with which services are developed (Polydorou, 1998). Therefore, there is a need to capture accurately the customer demands, and create and introduce rapidly the pertinent services, in a way that they retain certain desired quality characteristics despite their sophistication. A SCE should support the service developer in addressing these requirements.

Despite their significance, SCEs are not perceived in the same manner by all service developers. The term SCE is rather abstract and is defined / interpreted in various ways by different people. However, what is undeniable (by simply considering its name) is that service creation activities fall into the scope of a SCE. Service creation is the first stage of the service life-cycle, refers to the transformation of the often vague and imprecise requirements for a new service into code that implements the required service, and includes all the activities that are necessary in order to create a new service, either from other existing services / components or starting anew.

In this context, and taking into account that the service creation process should be supported by a suitable methodology, a SCE is considered as a logical framework incorporating a collection of appropriate, carefully designed and tested, customisable, and user-friendly software tools (together with a reuse infrastructure) that are used according to the service development methodology with the aim to assist the service developer(s) when applying the methodology by automating and simplifying as much as possible the service creation process, and by facilitating consistency and verification checks. This definition of a SCE emphasises that a SCE is actually a development environment aiming to support (in the most general case) teams of service developers working towards a common goal (the development of a specific telematic service) using shared information and a number of software tools.

According to this definition, a SCE is in close cooperation with the service development methodology. Additionally, the SCE definition reveals that the two most important constituent parts of a SCE are:

- A set of software tools ranging from GUI-driven rapid prototyping tools and high level (specification) languages for the design of the service logic to powerful code generators for translating the high level design into the corresponding service code. For this purpose, various existing software technologies and tools can be used, depending on their commercial availability, their compatibility, and their ability to interact in an integrated manner (Adamopoulos, 2002).
- A reuse infrastructure supporting reuse at different levels of abstraction using appropriate notation and modelling constructs, and possibly code written in a certain programming language. This infrastructure includes mainly a service component library that provides reusable service components, which can be used for the construction of telematic services (Polydorou. 1998)(RACE, 1995). For structuring a service component library and starting to develop it, the information and computational modelling

concepts and guidelines of an architectural framework for services (like OSA and TINA-C) should be considered.

The value of a SCE is increased as it is used for supporting the development of a variety of services. During this process, needs regarding software tools are identified and possibly satisfied, the service component library is enriched with new service components, and service developers gain important related experience. The result is a more advanced SCE with improved effectiveness. However, the success of a SCE depends also on more abstract factors. More specifically, since a SCE is used by service developers working in an organisation / enterprise (e.g. a service provider), it is influenced by the culture and general philosophy of the organisation regarding service development and the related technologies. It is also subject to the limitations of the organisation in terms of tool and personnel support.

Although the definition of a SCE is relatively simple and clear, constructing an effective SCE is a challenging task. For this reason, considerable research effort is being devoted to the development of SCEs and the examination of their structure (ACTS, 1997)(ACTS,1999) (Polydorou, 1998) (Adamopoulos, 2002). The main related attempts will be briefly presented in a critical manner in the following paragraphs, in order to facilitate the reasoning about the purpose and the structure of the SCE that should be used together with the proposed methodology.

SCEs initially appeared in conjunction with the technology of IN. Legacy IN systems usually encompassed powerful SCEs, where the service designer had only to select icons (representing predefined primitive components that rarely corresponded to standardised SIBs), interconnect them to each other, and fill-in some dialogue-based forms to parameterise them. This procedure was an easy one and the work of the service designer was simplified. However, these systems were closed, not able to seamlessly embed user-written code in order to enhance the functionality of the services. Hence, the creation of more complex services was tricky and demanding. With the progress of IN standardisation this problem was largely overcome and SCEs were introduced enabling the quick customisation or creation of IN services out of predefined open components (SIBs). Thus, the existing commercially available SCEs speed up the design of the IN service logic (even in the case of complex services), but the whole process is still relatively long. This is mainly due to the very limited support for specification, validation, and testing (especially early in the development process) provided by the current SCEs. Therefore, there is a need for adequate support by appropriate software tools.

The functionality and role of SCEs regarding the technology of IN was significantly extended by the Eurescom project P103 “Evolution of the Intelligent Network”. A service creation process, which combined the Object Oriented Role Analysis Method (OORAM), Message Sequence Charts (MSCs), and the Specification and Description Language (SDL) accompanied the SCE that was developed by this project. It contained several role models (which are units of modularity in OORAM) for the various service creation activities, without restricting the way they may be combined, and a “service constituent storage”, storing service constituents at various abstraction levels for reuse purposes and allowing their easy retrieval and maintenance. Rather than defining a new storage model, it has been decided to adapt an existing model developed within the RACE project SCORE (RACE, 1995) (see below).

In parallel with project P103, research regarding SCEs was also taken place in the context of the OSA architectural framework by the RACE project SCORE (Service Creation in an Object-oriented Reuse Environment). SCORE developed a service creation process model, which is generic, as it specifies roles, activities and associated tools, but it does not specify a particular process or temporal interconnection. This is left to be done by the enterprise developing and deploying services according to its specific needs. SCORE adopts a very broad view of what an SCE is, and after identifying the most important requirements that should be met by SCEs, considers the service creation process model as a generic SCE. Thus, a particular SCE (dedicated SCE) is an instantiation of the service creation process model, in which the particular tools and methods appropriate to the type of service being created, and to the enterprise doing the creation, are brought together.

Finally, the most recent detailed and systematic examination of SCEs for component-based service creation was conducted by the ACTS projects SCREEN (Service CReation Engineering ENvironment) (ACTS, 1999) and TOSCA (TINA Open Service Creation Architecture) (ACTS, 1997). The related results are in alignment with the architectural framework of TINA-C and were tested in several subsequent projects. More specifically, the main objective of SCREEN was to consolidate and extend existing and emerging technologies in order to produce a seamless tool-supported approach to component based service creation (Polydorou, 1998). Therefore, it developed an open, multi-vendor SCE, which consists of service logic development tools that provide the necessary means (software technologies) for the design, specification and implementation of the service logic, a service component library that promotes and facilitates the reuse of service components, and QoS related facilities that enable cost-effective service provision at the appropriate quality levels. The operation of this SCE is supported by a service creation practice structured according to a number of phases (requirements gathering and analysis, service analysis, service design, validation and simulation, and DPE targeting).

TOSCA was primarily focused on the development of a SCE capable of supporting the rapid (automatic) creation and validation of telecommunications services in an effective manner. The TOSCA approach assumes that for a certain class of service, a flexible and reliable object-oriented software framework based on TINA-C principles is developed (Lodge, 1999). A framework is a set of service components that can be used to build a large number of standard and customised services. It may be specialised through inheritance and / or delegation to acquire service-specific behaviour and represent the logic of a particular service. Framework specialisation is carried out through the use of a paradigm tool. This is a graphical CASE (Computer-Aided Software Engineering) tool that allows the service designers to specify the behaviour of a service at an abstract level, and then automatically creates the new service by appropriately customising a framework with the specific service behaviour. Thus, the TOSCA SCE consists of a framework and a corresponding paradigm tool supplemented with service validation tools (ACTS, 1997). Recognising the fact that the development of frameworks is a considerably more complex task than the development of services, TOSCA has also developed a process architecture for the development and use of frameworks in service creation (Lodge, 1999).

The SCREEN and TOSCA approaches to SCEs are complementary. Taking into account that the resources required to develop a framework are far greater than the resources required to create one of the services in the class covered by the framework, the SCREEN approach should be used for the initial development of new types of services. Then, if there is a high confidence level that there will be a need for similar services at some point in the future, an object-oriented framework should be developed (using components from the original service) to facilitate the creation of a class of similar services using the TOSCA approach.

A detailed comparison between different existing SCEs that are applicable to TINA-C is not attempted, because it is argued that every SCE can be sufficient and effective in a certain service development context. Thus, the importance of supporting efficiently the entire service creation process is highlighted, because the characteristics of a specific service development methodology determine the kind of support that the service developer(s) expect from an SCE and thus (should) affect significantly its structure and its constituent parts. In this context, the success of a SCE is measured against its ability to facilitate the service developer(s) when applying each of the phases of a particular service development methodology, and is determined by the quality and the effectiveness of the corresponding methodology.

The construction and use of an SCE, without having a specific service development methodology in mind, is not suggested, because it cannot offer any kind of guarantee regarding the result of the service creation process. Only an experienced service developer will manage to design and implement a successful telematic service (i.e. according to user requirements), without any methodological support and only the use of a SCE and some abstract guidelines. Furthermore, his / her attempt will require an undefined and impossible to predict amount of time and will involve a significant amount of ad hoc decisions on a variety of matters, preventing in that way any kind of project management activity (including the possible cooperation with other developers) and seriously jeopardising the maintainability of the service. Finally, there is always the danger of a service developer who, mislead by the capabilities of the (software tools of the) SCE, proceeds to activities which either destroys his / her focus on the real service development problems, or directs his / her interest away from considering the real user requirements, with obvious negative effects on both the service creation process and on the resultant telematic service.

An intermediate approach is followed by the SCREEN project, which recognises the importance of specifying a service creation methodology when developing and using an SCE. Thus, SCREEN provides (a kind of) a methodology addressing service creation matters. However, this methodology is abstract. It has a rather obvious structure, which is not elaborated, and acts more like a loosely-coupled framework for fitting in several concepts, practices and technologies. Thus, SCREEN fails to provide sufficient methodological support regarding the service creation process and thus the use of the SCE that it prescribes does not deliver the anticipated benefits for reasons already explained previously.

The same remark applies also to the TOSCA approach, which although it has a much wider scope than SCREEN, it does not specify a service development methodology. Furthermore, TOSCA underestimates the need for such a methodology by adopting a paradigm based approach to service creation and a navigation and parameterisation approach to service customisation (ACTS, 1997). However, composing telecommunications services from reusable service components without the support of an appropriate service creation methodology has important drawbacks. Additionally, in the case of the TOSCA paradigm based approach, each TOSCA SCE only allows the service developer to create services within some specific set of possible services (the class of services covered by the framework). This set may be large but it is inevitably restricted by both the functionality of the framework and the nature of the paradigm chosen. Similar limitations exist also in the case of navigation and parameterisation, as the service type required must be already available if the service developer is to be able to select it. Therefore, for the creation of any arbitrary service a process of software development in the form of a service creation methodology must be followed.

Therefore, the purpose of the SCE is to increase the applicability of the service development methodology, by efficiently facilitating the service developer(s), and not to substitute it. For this reason, the SCE is proposed to be constructed (or customised) by the service developer(s), prior to the start of the service development project, according to:

- The exact characteristics and nature of the service development methodology that they plan to use.
- Their experience regarding the use of appropriate software tools.
- The requirements (the "design culture") of the organisation / enterprise that they work for.
- The requirements of the organisation / enterprise that will use (and possibly maintain) the telematic service that will be developed.
- Any special requirements that the telematic service under development may impose.

In this manner, the maximum flexibility is achieved and the service creation process is facilitated in the best possible way by the SCE. It is evident that this approach can incorporate (after appropriate customisation) the main findings regarding SCEs of both SCREEN and TOSCA, as long as they do not contradict the service development methodology that is used. This remark applies especially to the reuse infrastructure, which is less affected by the methodology.

5 CONCLUDING REMARKS

Even IN-augmented PSTN, fails to meet the demand for a great diversity of broadband, multi-media, and multi-party telecommunications services with enriched functionality. Such services require a more flexible service provisioning approach, in which flexibility should be pursued even if it is not advantageous from a performance point of view, and in which service - transport network dependencies are eradicated. DPEs can satisfy this need by permitting very complex signalling interactions, abstracting over the use of low level signalling protocols without semantic loss. Under these conditions telematic services can be considered as complex distributed software applications designed and implemented as a set of interacting service components operating upon a DPE. This complexity together with the generic character of modern networks, which are not bound to a specific "service paradigm", highlight the necessity for a service creation methodology that will bridge the gap between the initial conception of a service and its actual implementation.

Recognising this need, this paper proposes such a methodology with an incremental and iterative character that should be used together with a suitably structured SCE. The paper argues that the SCE should be in close cooperation with the service creation methodology and aims to increase its applicability and not to substitute it. Therefore, the SCE is proposed to be constructed / customised by the service developer(s), prior to the start of a specific service development project.

REFERENCES

ACTS Project AC237 (TOSCA), 1997. *Service Creation: the TOSCA Paradigm and Framework Approach*, CEC Deliverable AC237/BT/DS/P/019/B1.

ACTS Project AC227 (SCREEN), 1999. *Final Report*, CEC Deliverable D12, March 1999.

Adamopoulos, D.X., Pavlou, G., Papandreou, C.A., 2002. Advanced Service Creation Using Distributed Object Technology. In *IEEE Communications Magazine*, Vol. 40, No. 3, pp. 146-154.

Lodge, F., Kimbler, K., Hubert, M., 1999. Alignment of the TOSCA and SCREEN Approaches to Service Creation. In *Proceedings of IS&N '99*, LNCS 1597, Springer-Verlag, Berlin, pp. 277-290.

Polydorou, N.D., Liossis, N.I., Tzifa, E.C., Demestichas, P.P., Anagnostou, M.E., 1998. Efficient Creation and Deployment of Telecommunication Services in Heterogeneous Distributed Processing Environments. In *Proceedings of IEEE/IEE ICT '98*, Vol. IV, Greece, pp. 336-340.

RACE Project R2017 (SCORE), 1995. *The SCORE Component Model and its Framework*, Deliverable D105, Vol. II.