A Methodology for the Development of New Telecommunications Services

DIONISIS X. ADAMOPOULOS
Centre for Communication Systems Research
School of Elec. Eng., IT and Mathematics
University of Surrey
Guildford GU2 5XH
ENGLAND

GEORGE PAVLOU
Centre for Communication Systems Research
School of Elec. Eng., IT and Mathematics
University of Surrey, England
Guildford GU2 5XH
ENGLAND

CONSTANTINE A. PAPANDREOU
Hellenic Telecommunications Organisation (OTE)
17 Kalliga Street, GR-114 73 Athens
GREECE

Abstract: - The telecommunications industry is currently facing a growing need of making telecommunications services more versatile, easier to develop, interoperable, consistent, manageable, and independent of the underlying network. Additionally, the demand for new sophisticated telecommunications services with multimedia characteristics is increasing. These services require more flexible access, management, and charging mechanisms. Therefore, it is necessary to assist, in a systematic manner, telecommunications service designers in the development of new services. In this paper, after a brief examination of important service engineering matters, a service creation methodology is proposed and presented focusing on its essential characteristics. A possible enhancement of this methodology through the use of design patterns and frameworks is then considered.

Key - Words: - Service development, service creation, new telecommunications services, TINA-C, service engineering.
IMACS/IEEE CSCC'99 Proceedings, Pages:5061-5065

1 Introduction

The telecommunications market is currently characterised by a fast evolving technology, a multiplicity of cooperating and competing providers, and a growing demand for a great variety of new specialised multimedia services. Recent developments in broadband networks and distributed computing have increased the feasibility of creating telecommunications services as distributed application programs in open distributed systems. These systems are characterised by the heterogeneity of the underlying computing and networking technology. In order to deal with this heterogeneity many distributed platforms (generically called Distributed Processing Environments, DPEs) have been developed.

To meet the needs of future telecommunications in an integrated and effective manner, various architectural frameworks are being developed and applied (TINA-C, OSA), that provide the means to build services and a service support environment One of the major goals of these architectural frameworks is to assist and constrain designers in the complex process of service creation and design.

This paper considers important issues that underpin the service creation process in a highly competitive environment of service provisioning. For this reason, a Telecommunications Information Networking Architecture Consortium (TINA-C) conformant 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.

2 The Need for an Integrated and Systematic Approach

An architectural framework 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, TINA-C is more descriptive rather than prescriptive, and its application can be a complex task [6]. Furthermore, there seems to be no end to the emergence of new services, each requiring 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.

The basic factors that shape this challenge are addressed by the discipline of integrated service engineering, which includes the technologies and engineering processes required to define, design, implement, test, deploy, maintain, and manage telecommunications services. The concept of the service life-cycle has a central role in integrated service engineering. All services go through a service life-cycle, which contains descriptions of activities, in the form of an ordered collection of processes or steps, that are required to support the development, the operation, and the maintenance of a service [7].

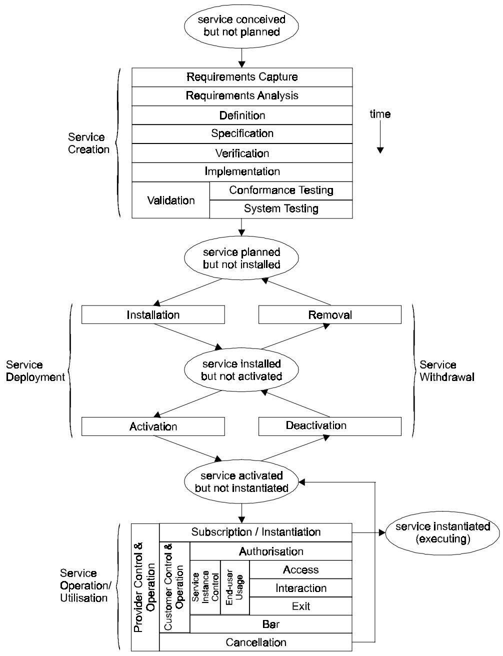

Fig. 1: The service life-cycle.

Fig. 1 depicts a graphical representation of a service life-cycle, which is an enriched variation of the TINA life-cycle model. The rectangles are the actions / phases, while the ellipses are the main states that a service goes through (conceived but not planned, planned but not installed, installed but not activated, activated but not instantiated, and instantiated (executing)).

Among all the stages of the service life-cycle of Fig. 1, in TINA-C, service creation is one of the most abstract and general, since it does not provide many guidelines on how to structure each of its phases. Furthermore, it is also one of the most important as it determines the efficiency with which the services will be developed, and thus the success of service providers in a highly competitive market.

In order to meet these challenges, a service creation methodology is proposed to enable TINA-C to satisfy the required expectations on long-term efficiency of service design, provision, and management. 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 TINA-C compliant DPE wherein the service will be deployed, facilitates the design and implementation of a TINA-C compliant service, which meets the desired requirements by making use of the service independent features to the larger possible extent [3] [9].

3 The Proposed Methodology

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

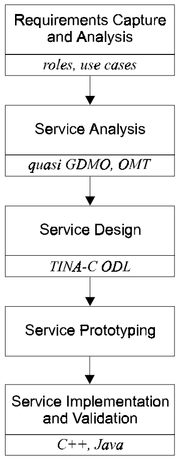

Fig. 2: The proposed methodology for the development of telematic services.

The proposed methodology contains a conceptual model of constructs and a series of guidelines, essential to the development of telematic services, and a set of partially ordered procedures suggesting the direction to proceed. An overview of the proposed service creation methodology can be seen in Fig. 2.

As can be seen from Fig. 2, the proposed methodology uses the service life-cycle of Fig. 1 as a roadmap. It has to be stressed that the proposed methodology does not imply a waterfall model. For simplicity, iterative aspects of the involved processes and feedback loops are not shown in Fig. 2 but they are intended. Furthermore, graphical and textual notations are proposed for almost all phases to improve the readability of the related results and ensure a level of formalism sufficient to prevent any ambiguity. In the following paragraphs each of the phases of the proposed methodology are examined.

3.1 Requirements Capture and Analysis

During this phase the service developer assembles, documents, and structures the requirements on the service (service needs) from the different stakeholders involved. The focus is on modelling the concepts that are visible at the service boundary, and thus the service logic is viewed as a black box.

One of the most important tasks pertaining requirements capture is the identification of the independent entities / actors, which are involved (by their collaboration) in the operation of the service within and across business administrative domain boundaries. These entities correspond to roles modelling a well defined grouping of functionality under control of a specific stakeholder [11].

A role can be either generic or specific. The main generic (business) roles and the (business) relationships between them are specified by the TINA Business Model [10]. Each generic role (consumer, retailer, broker, third party service provider, connectivity provider) corresponds to one or more specific roles.

TINA-C reference points are defined in relation to the (business) relationships they support, and, according to their functionality, can be divided into access segment and usage segment reference points. This segmentation of reference point definitions enables any inter-domain reference point to be defined with the minimum set of functionality needed for the (business) relationships being analysed.

In order to proceed to the service analysis phase, the reference point segments should be mapped to feature sets, which are groups of interfaces that expose restricted parts of an information model for manipulation or examination, define the details of interactions between components, and specify levels of functionality inside a service (e.g. basic or multiparty session control). Via the mapping to feature sets, the segments for any particular inter-domain reference point should determine the minimum set of service components (or related functionality) that would be required to conform to this reference point.

3.2 Service Analysis

The aim of this phase is to determine the functionality needed for satisfying the requirements that were identified in the previous phase, and to define the software architecture of the service implementation. For this reason, the focal point shifts from the service boundary to the internal service structure. The output of this phase is (mainly) the static view of the internal structure of the service.

The service analysis phase is the first phase of the service creation process, where the service is decomposed into constituent parts (Information Objects, IOs), with the appropriate relationships among them, in an attempt to gain an overall understanding of the service. High level Object Modelling Technique (OMT) diagrams are used to represent the main concepts of the service and their relationships. Additionally, analysis level Message Sequence Chart (MSC) diagrams model interaction among service parts. Alternatively, Unified Modelling Language (UML) static chart diagrams can be used to describe the static structure of the IOs, and UML statechart diagrams to specify the dynamic behaviour of significant IOs.

The OMT and MSC models defined so far provide a high level overview and a clear understanding of the service as a whole. They do not focus to the individual IOs. Consequently, a formal modelling syntax is needed that allows the semantics of the information model to be precisely captured and specified. Quasi GDMO (Guidelines for the Definition of Managed Objects) with GRM (General Relationship Model) are the formal notations used for this reason.

According to the TINA-C service architecture new services can be realised by enhancing already existing components (e.g. with the use of inheritance) or by defining new ones. TINA-C specifies a set of service independent components that support access session, generic service session, and communication session related functionality. These components can be considered as reusable units in the creation of new services. They may be used in a service implementation as they are, or as the basis for the construction of a service specific component [9].

The exploitation of the TINA-C service independent components in the service analysis phase, and in subsequent phases (depending on the nature of the available component libraries), begins with the selection and reuse of the appropriate service independent functionality. Then, the service dependent segment is developed, by exploiting as much as possible the service independent segment. Finally, the two segments are integrated. This process can be expressed with the following series of steps (fine tuning implies a feedback loop):

- Configure the access session related segment:
  - Select the access session related functionality.
  - Customise (if necessary) the selected access session related functionality.
- Configure the service session related segment:
  - Select the service generic functionality.
  - Customise (if necessary) the selected service generic functionality.
  - Determine the service specific functionality.
  - Develop the service specific functionality.

- Fine tune the relations between the service generic and the service specific part.
- Fine tune the relations between the access session and the service session segment.
- Configure the communication session related functionality:
  - Select the communication session related functionality.
  - Customise (if necessary) the selected communication session related functionality.
- Fine tune the relations between the service session and the communication session part.
- Integrate all three segments (access, service, and communication session).
- Prepare the end user system:
  - Develop an end-point of the service session.
  - Develop / customise an end-point of the access session.

3.3 Service Design

During this phase the service developer defines the interfaces and the behaviour of the IOs that were identified in the service analysis phase, and structures the service in terms of interacting computational objects (service components). The output of this phase is the dynamic view of the internal structure of the service.

As a first step in this phase, the IOs that comprise the service are considered as potential candidates for Computational Objects (COs). In many cases, IOs are mapped to one corresponding CO encapsulating the information defined by the IO and providing an operational interface to access the information. However, the mapping between IOs and COs is not necessarily one to one.

COs are specified using the TINA-C Object Definition Language (ODL). Additionally, UML collaboration, sequence, activity, and component diagrams can be used. While the collaboration diagram shows the interactions among object instances and their links to each other, the sequence diagram describes object interactions arranged in a time sequence. The activity diagram allows to specify in which order activities (such as operations provided by a CO interface) have to be executed. Finally, the component diagram shows the organisations and dependencies among components.

3.4 Service Prototyping

In this phase a prototype of the service is being developed utilising special software tools and used to determine the equivalence between the service specification (the result of the service design phase) and the initial requirements (the result of the requirements capture and analysis phase). In the case of differences adaptations have to be made.

By developing a service prototype uncertainties about the service can be resolved. Short-term additional costs can result in long-term savings as requirements and design decisions are clarified during the prototyping phase.

The service prototype can be created with the intention to be discarded after its evaluation ("throw-away" prototype). An alternative approach is to use the prototype as the basis for the subsequent service implementation phase (exploratory prototyping). In this scenario, the telematic service "evolves" by an iterative process. The management and control of this process is important to ensure that compromises are not made and the process of repeated iteration does not go on too long [1].

3.5 Service Implementation and Validation

During this phase an implementation is generated from the (verified) service specifications (possibly using the service prototype developed in the previous phase), and the deployability of the overall implementation on a TINA-C compliant DPE is examined (DPE targeting).

The engineering representation of a CO (using an object-oriented programming language like C++ or Java) is called an engineering Computational Object (eCO). The mapping between COs and their eCOs is one to one: no eCO represents a composition of COs nor is a CO represented by more than one eCOs. The interfaces of an eCO represent the interfaces of its corresponding CO [4].

Furthermore, validation takes also place in this phase by comparing the developed service software against the service specifications produced at the service design phase. This activity can be subdivided into the following two subactivities:

- *Conformance testing:* It involves checking the implementation for conformance to architectural rules and standards used in the service design.
- *System testing:* It comprises the testing of service software in a (possible) operational environment.

In this phase, the UML component and deployment diagrams can be used to define mechanisms and functions required to support distributed interactions between service objects in the target DPE. In general, the use of the same UML notation in almost all the phases of the methodology, has the advantage of making both the service description more coherent and the process of proceeding from one phase to another more natural and efficient. For this reason, TINA-C could also consider UML in a future version of its computing and service architectures.

4 Exploitation of Design Patterns and Frameworks

A typical telematic service is a large scale system. For this reason, the successful application of the proposed service creation methodology is a rather complicated task, where the effective and efficient communication of architectural knowledge between the service designers and developers is of great importance. To facilitate this communication the exploitation of design patterns and frameworks in the service engineering area is suggested.

In the case of TINA-C, design patterns can be defined by identifying groups of interworking service objects, where every group is characterised by a micro-architecture that determines the way the objects interact to provide a solution for the specific aspects of a subproblem that arises during the development of a telematic service. Furthermore, a framework can be defined as the overall architecture, which specifies how the identified configurations of service objects can collaborate to implement a solution for the whole problem. Thus, a framework is a kind of construction kit for complete or semicomplete telematic services. It has to be complemented and customised using inheritance techniques [2].

The introduction of design patterns and frameworks in the proposed methodology implies the establishment of a common vocabulary and the definition of common design structures for all persons involved. They assist to reduce the scope of the problem solving process in the case of service creation, because they support the identification of similar problems and similar solutions. However, design patterns and frameworks are abstract concepts. There is no guarantee that their usage will lead to design reusability, design portability, and abstract customisability. Furthermore, good design patterns and frameworks, like good inheritance hierarchies, can not be invented in an easy way. They have to be chosen and designed very carefully [5].

6 Conclusions

Explosive increase in service variety as well as in globalisation and customisation of services produce ever increasing pressures for the efficient support of the service engineering activities in an environment open to changes in market and technology. For this reason, in this paper, the key features that should characterise an advanced service creation and design methodology were presented and examined.

The proposed methodology is being applied to the development of a multimedia conferencing service for education and training [8]. The results obtained so far provide confidence that this methodology is useful for the description and development of (complex) new telecommunications services. However, further experience on the application of the methodology is necessary, together with the establishment of a collection of evaluation criteria and metrics for more comprehensive verification procedures.

Under these conditions the proposed service creation methodology is expected to enrich TINA-C and enforce it to pave the way towards an open integrated broadband telematic infrastructure populated by a virtually limitless variety of telematic services.

References:

[1] Adamopoulos, D., Haramis, G., Papandreou, C., Rapid Prototyping of New Telecommunications Services: A Procedural Approach, *Computer Communications*, Vol. 21, 1998, pp. 211-219.

[2] Eckert, K., Schoo, P., Engineering Frameworks: A Prerequisite for the Design and Implementation of Distributed Enterprise Objects, *Proceedings of EDOC '97*, October 1997, pp. 170-181.

[3] Efremidis, S., et al., TINA-oriented Service Engineering Support to Service Composition and Federation, *Proceedings of IS&N '98*, LNCS, Vol. 1430, Springer-Verlag, 1998, pp. 409-422.

[4] Kelly, E., Mercouroff, N., Graubmann, P., TINA-C DPE Architecture and Tools, *Proceedings of TINA '95*, February 1995, pp. 39-54.

[5] Koerner, E., Patterns for Constructing CSCW Applications in TINA, *Proceedimgs of IDMS '97*, LNCS, Vol. 1309, Springer-Verlag, 1997, pp. 322-329.

[6] Magedanz, T., TINA - Architectural Basis for Future Telecommunications Services, *Computer Communications*, Vol. 20, 1997, pp. 233-245.

[7] Mudhar, P., A Service Creation Environment for a Future Intelligent Network, *Proceedimgs of IS&N '94*, LNCS, Vol. 851, Springer-Verlag, 1994, pp. 333-342.

[8] Papandreou, C.A., Adamopoulos, D.X., Design of an Interactive Teletraining System, *British Telecommunications Engineering*, Vol. 17, Part 2, August 1998, pp. 175-181.

[9] Polydorou, N.D., et al., Efficient Creation and Deployment of Telecommunication Services in Heterogeneous Distributed Processing Environments, *Proceedings of ICT '98*, Vol. IV, June 1998, pp. 336-340.

[10] TINA-C, *TINA Business Model and Reference Points 4.0*, Baseline document, May 1997.

[11] Wind, B., et al., Enhancing the TINA Architectural Framework for Personal Mobility Support, *Proceedings of IS&N '98*, LNCS, Vol. 1430, Springer-Verlag, 1998, pp. 233-248.