Advanced Service Creation Using Distributed Object Technology

Dionisis X. Adamopoulos and George Pavlou, University of Surrey, England
Constantine A. Papandreou, Hellenic Telecommunications Organization (OTE)

ABSTRACT

Rapid technological developments are taking place in computing and telecommunications that provide a wide range of opportunities for the delivery of new advanced multimedia services (telematic services). In this context *information networking* has a central role, and *telecommunications service engineering* emerges as an important new scientific discipline. This article examines important issues that underpin the creation of telematic services in a highly competitive environment of service provisioning. The starting point is the proposal of a *telecommunications service engineering framework* consisting of a service development methodology, a service creation environment, and a service support environment, that considers telematic services as distributed object-oriented applications operating on distributed object platforms. Then the article focuses on the presentation and examination of a *complete methodology for the development of telematic services*, that "covers" in a systematic and structured manner the entire service creation process. Finally, the proposed methodology is *validated* by applying it to the design and development of a complex representative telematic service.

INTRODUCTION

The telecommunications industry is currently undergoing a fundamental restructuring as the evolving synergy between computing and telecommunications technology, termed *telematics*, is gradually gaining momentum. This evolution and diversification is being driven mainly by liberalization, increasing competition in the marketplace, technological advancements, and demands from all customer segments for an increasingly sophisticated portfolio of telecommunications services tailored to specific needs. It is expected that the forthcoming integrated (fixed and mobile) broadband networks will be openly available to existing and new service providers, constituting a worldwide common shared communications platform for a multitude of new advanced telecommunications services (telematic services) [1]. The proliferation of these services will lead eventually to the transformation of the global information infrastructure to an *open services market* fueled by deregulation, strategic partnerships, and joint interoperability activities [2].

A direct result of this trend is the dramatic increase in the number, variety, and sophistication of *telecommunications services* telecommunication companies offer in an attempt to satisfy the high and continuously expanding customer demand for powerful communication and information capabilities. In this context *information networking* is gradually gaining momentum, formulating an open market of new telecommunications services where the vision is "information any time, at any place, in any form." Under these conditions *software* has a central role. The traditional method of telecommunications service development and provision is not possible to meet the increasing demands of a competitive, highly dynamic telecommunications services market. Such a market requires increased "intelligence" of telecommunication networks, provided by software whose importance is constantly rising [3]. Based on the principles of distributed computing systems, an information network is realized as a distributed processing system in which the telecommunications services can be regarded as distributed software applications [4].

In the light of these challenges, this article, after a brief introduction to the scientific discipline of telecommunications service engineering, proposes a framework for the efficient structuring and conducting of service engineering activities. The core of the article is devoted to the examination of a novel service creation methodology, which is the most important constituent part of the proposed framework, by focusing on its essential characteristics, with the intention of assisting service developers, in a structured and systematic manner, during the entire service development process. The validation of this methodology is also considered with the aim to provide tangible evidence about its correctness, its efficiency, and its true practical value.

The Emergence of Telecommunications Service Engineering

In order to derive a viable paradigm for the broadband information highway of the future, *telecommunications service engineering* (or simply *service engineering*) should emerge as a separate scientific discipline of strategic importance. Service engineering can be defined as the discipline addressing the technologies and engineering processes required to define, design, implement, test, verify, validate, deploy, maintain, and manage telematic services that meet user needs in current or future networks. The concepts, principles, and rules of service engineering were initially borrowed from software engineering, although there is a clear distinction between the two disciplines. However, in order to handle the large size of possible services, their open-endedness, and the variety of systems that will have to interwork, service engineering gradually incorporated a vast number of concepts already defined and developed in the fields of both information technology and telecommunications. Its main objective is to ensure the introduction of new and enhanced services and their management in a fast and efficient manner.

A significant attempt to promote service engineering principles originated by the *Telecommunications Information Networking Architecture Consortium* (TINA-C). The main objective of TINA-C is to define, validate, and promote an open, innovative, and coherent architectural framework (a long-term architecture for telematic services), based on object-oriented design principles, that would address in an integrated manner service control and service management [5]. The TINA-C framework is already in a *mature* state and is gaining acceptance, through validation activities and the examination of interworking concepts and migration strategies from existing technologies. In general, there is evidence that TINA-C is a very interesting and promising solution for the middle and long term, and it is an *essential evolutionary step* toward the optimum (re)structuring of the emerging multiservice networks (MSNs).

The Proposed Integrated Approach

Service engineering activities, in order to retain their usefulness and fulfill the emerging increased expectations regarding their value and impact, have to transform and adapt to the new conditions that shape the telecommunications world of the 21st century. The corresponding evolution process is characterized by a number of *requirements* that service engineering activities should satisfy in the new era [6]. More specifically, service engineering activities should support:

- The efficient and effective development of telecommunications services
- The successful application in a telecommunications context of computer science findings
- The reduction of complexity and increase of efficiency
- The efficient automation of the service creation process
- The development of a rich variety of telecommunications services
- The exploitation of new and/or emerging communication concepts
- The representation, processing, management, and transmission of all the basic information types
- Precise service semantics
- Reusability at different abstraction levels
- The flexible management of services
- The interoperability of services
- The accommodation of relevant standards
- The efficient and accurate handling of service interactions
- Openness to all types of potential end users of a service
- Openness to change of service software and hardware
- The accommodation of legacy telecommunications services and systems

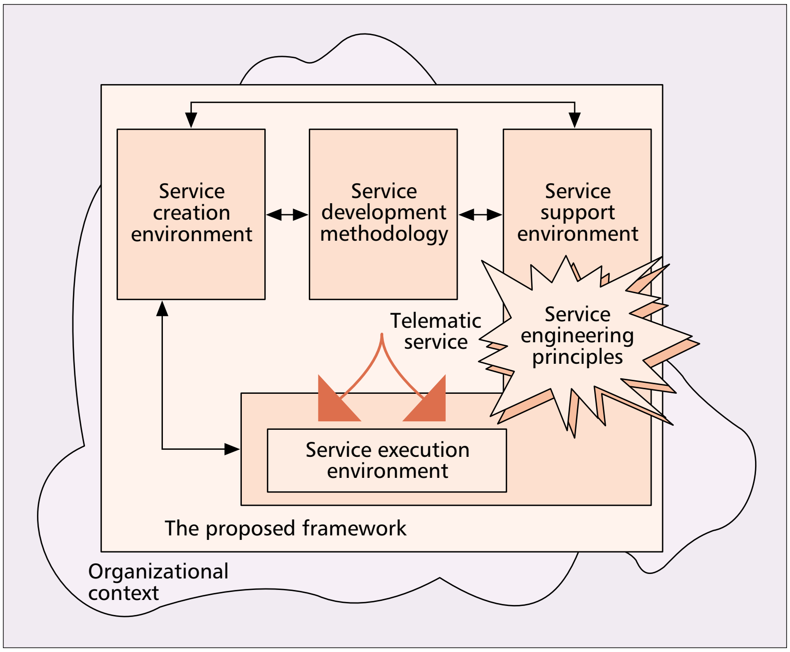


Figure 1. *The proposed telecommunications service engineering framework.*

In an attempt to revitalize telecommunications service engineering and prepare it for the crucial role it is anticipated to have in the new emerging telecommunications environment, the telecommunications service engineering framework of Fig. 1 is proposed; it is argued that its adoption will enable (the various) service engineering activities to satisfy *all* the previously identified requirements (*not* just a subset of them). As can be seen from Fig. 1 the proposed framework is placed inside an *organizational context* in order to signify that service engineering activities are normally performed by service developers or service designers working for various organizations or enterprises. Therefore, it is influenced by their knowledge, problem solving

The creation of telecommunications services within an open environment is a highly complex activity. This complexity stems not only from the technical nature of the tasks involved, but also from the number of the participating actors and the variety in their roles, concerns, and skills.

attitude, and experience, and also by the more general telecommunications and IT strategic orientation of the organization/enterprise for which they work.

The main constituent parts of the proposed telecommunications service engineering framework, which are depicted in Fig. 1, are:

A service development methodology: It is a methodology that guides service developers in a systematic and structured way during the entire process of service creation.

A service creation environment (SCE): It is actually a collection of software tools (together with a reuse infrastructure) used according to the service development methodology with the aim to assist the service developer(s) when applying the service development methodology by automating and simplifying as much as possible the service creation process, and facilitating consistency and verification checks.

A service support environment: It is an environment aiming to facilitate both the development of telematic services (in cooperation with the service development methodology and SCE) and their execution under real conditions. It consists mainly of:

- *Service engineering principles*: These are concepts, guidelines, practices, and (in general) mental constructs that are applicable to service engineering activities.
- *A service execution environment*: This encompasses the necessary computing and network infrastructure and the appropriate ancillary software (e.g., operating systems, database management systems) needed for and during the execution of a telematic service. However, the most important part of the service execution environment is the distributed processing environment (DPE), which abstracts over all its constituent parts and greatly reduces the effort needed for implementation of a telematic service.

The service development methodology is the main constituent part of the proposed telecommunications service engineering framework since it presents increased research interest and practical value. The creation of telecommunications services within an open environment is a highly complex activity. This complexity stems not only from the technical nature of the tasks involved, but also from the number of participating actors and the variety in their roles, concerns, and skills. Therefore, there is a need to support the *complex service creation process* in order to ensure that resulting services actually perform as planned and as required by customers and service providers. A methodology is an important part of such an attempt, since it provides a systematic and structured base for the flexible and efficient management of the development of telecommunications services. Thus, the rest of this article, in order to structure and control the service development process from requirements capture and analysis to service implementation and testing, reduce the inherent complexity, and ensure the thorough compatibility among the many involved tasks, proposes and examines a novel TINA-C-conformant object-oriented *service creation methodology*.

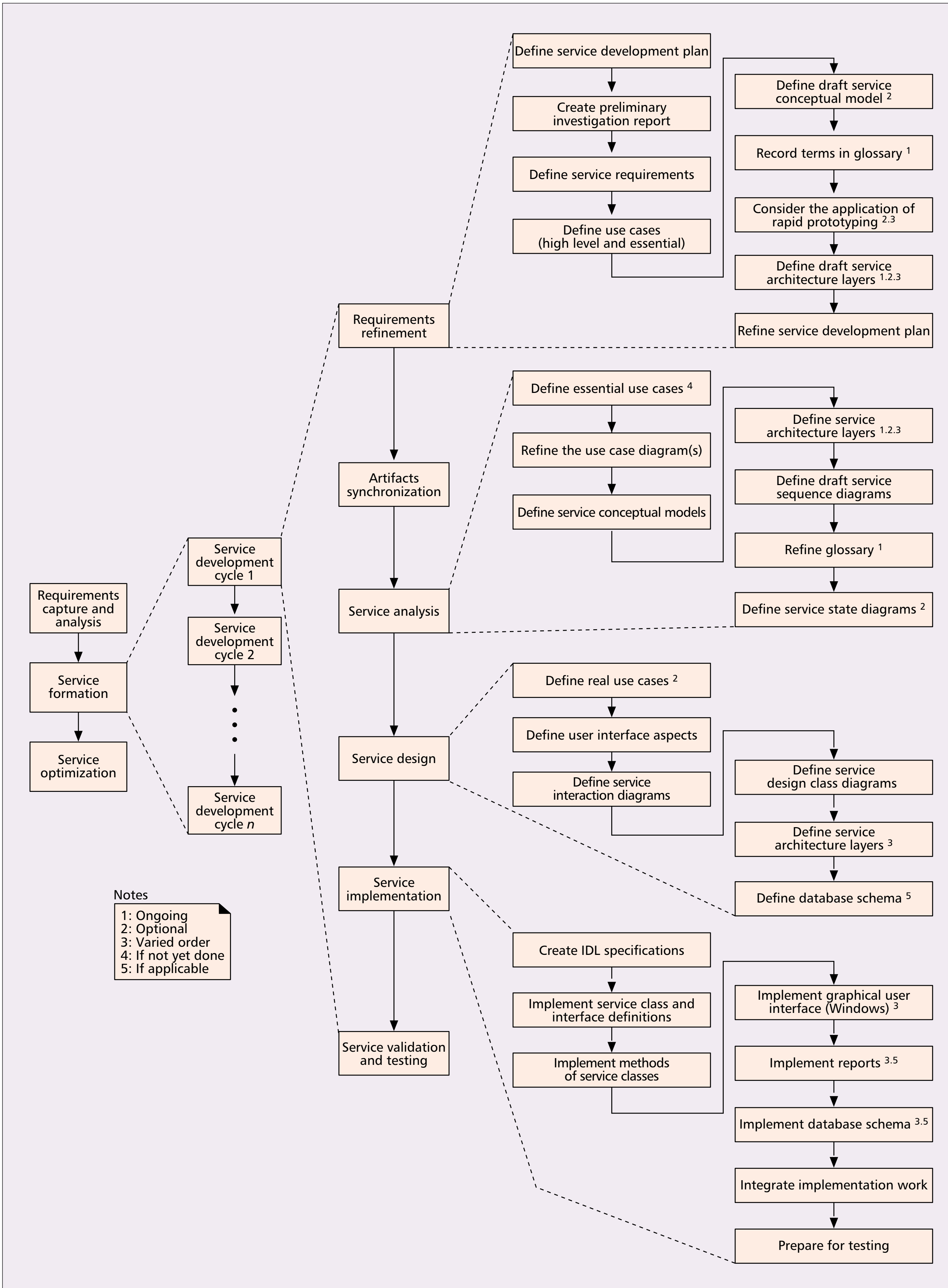
A Methodology for the Development of Telematic Services

Among all the stages of the service life cycle *service creation* is one of the most abstract and general, since there are not many detailed guidelines (advisory statements) available on how to structure each of its phases. Furthermore, it is also one of the most important since 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 this challenge, 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 way without making quality compromises in an open deregulated multiprovider telecommunications marketplace.

An overview of the proposed service creation methodology depicting the activities of its main phases can be seen in Fig. 2. Although this figure suggests a linear order of artifact creation, this is not strictly the case. Some artifacts may be created in parallel. 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 (which are actually subphases of the service formation phase). The telematic service grows incrementally as each cycle is completed.

As can be seen from Fig. 2, the proposed methodology does not imply a waterfall model in which each activity is done once for the entire set of service requirements, because the traditional waterfall model for software development with its sequential phases is inadequate to support the development of telecommunications services. Furthermore, graphical and textual notations are proposed in each phase in order to improve the readability of the related results and ensure a level of formalism sufficient to prevent any ambiguity. The proposed use of the Unified Modeling Language (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 [7]. For this reason, TINA-C could also consider UML in a future version of its computing and service architectures.

According to Fig. 2 the main phases of the proposed methodology are the following.



■ **Figure 2.** *An overview of the proposed service development methodology.*

The service analysis phase is the first phase of the service creation process where the telematic service is decomposed into its constituent parts, with the appropriate relationships among them, in an attempt to gain an overall understanding of the service.

REQUIREMENTS CAPTURE AND ANALYSIS PHASE

During this phase the service developer assembles, documents, and structures the requirements on the service from the different stakeholders involved. The focus is on modeling the concepts that are visible at the service boundary; thus, the service logic is viewed as a black box. The primary goal of this phase is to identify what functionality is really needed to include in the telematic service, and document it in a form that is easily understandable and unambiguous.

In order to improve understanding of the service requirements, use cases are created. They describe the sequence of events generated by an actor using a telematic service to complete a specific service process. Use cases may be expressed with varying degrees of detail (leading to high-level and expanded use cases) and commitment to design decisions (leading to essential and real use cases).

SERVICE ANALYSIS PHASE

The aim of this phase is to determine the functionality needed to satisfy the service requirements 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 service analysis phase is the first phase of the service creation process where the telematic service is decomposed into its constituent parts, that is, service information objects (IOs) or service concepts, with the appropriate relationships among them, in an attempt to gain an overall understanding of the service. The resulting main *service conceptual models* are accompanied by a set of *ancillary service conceptual models*, which are deduced from the TINA-C service architecture. In UML, a service conceptual model can be illustrated with a set of static structure diagrams in which no operations are defined.

In order to better understand the service behavior, a service sequence diagram, which is a UML sequence diagram, is constructed for every use case, depicting the service events that external actors generate and their order. The behavior of a telematic service is further defined by service operation contracts, which describe the effect of service operations on the telematic service. They illustrate changes in the state of the overall telematic service when a service operation is invoked. UML contains support for defining service contracts by allowing the definition of pre- and post-conditions of service operations.

SERVICE DESIGN PHASE

During this phase the service developer defines the behavior of the service IOs that were identified in the service analysis phase and structures the telematic service in terms of interacting service computational objects (COs)/service components, which are distributable multiple interface service objects. They are the units of encapsulation and programming. While service IOs mainly explain how a service is defined, *service COs* reveal what actions have to be performed in order to execute the service. Therefore, the output of this phase is (mainly) a dynamic view of the internal structure of the telematic service.

After identifying the service COs, a service interaction diagram (which can be expressed as a UML collaboration or sequence diagram) is created for each service operation under development in the current service development cycle. Another important artifact created during service design is the service design class diagram, which illustrates the specifications for the software classes of a telematic service using a strict and very informative notation.

SERVICE IMPLEMENTATION PHASE

During this phase an implementation of the telematic service (service code) is generated from the service specifications, and the deployability of the overall implementation on a TINA-C-compliant DPE is examined (DPE targeting). It is assumed that at the beginning of this phase a specific (object-oriented) programming language (e.g., C++, Java) and a specific distributed object platform are chosen (e.g., Microsoft's DCOM, OMG's CORBA). The engineering representation of a service CO is called an *engineering computational object* (eCO). The mapping between service COs and their eCOs is one to one: no eCO represents a composition of service COs; nor is a service CO represented by more than one eCOs. The interfaces of an eCO represent the interfaces of its corresponding service CO, although there may be modifications depending on the exact characteristics of the selected DPE.

SERVICE VALIDATION AND TESTING PHASE

Validation takes place in this phase by comparing the developed service software against the service specifications produced at the service design phase and by ensuring that it meets the service requirements. *Testing* is proposed to be done using conventional testing practices (e.g., walkthroughs, condition and data flow testing, boundary value analysis), and it is evident that it can initiate reconsideration of the service code and possibly its modification.

With this phase a service development cycle ends, and another (depending on the exact nature of the specific service requirements) is ready to start. A significant strength of the iterative and incremental service development process adopted by the proposed methodology is that the results of a prior service development cycle can feed into the beginning of the next service development cycle. Thus, subsequent service analysis and service design results are continually being refined and informed by prior service implementation work. For example, when the code in cycle N deviates from the service design of cycle N (which it inevitably will), the final service design based on the implementation can be input into the service analysis and service design models of cycle $N + 1$. For this reason, as can be seen from Fig. 2, an early activity within a service development cycle is to refine the requirements and synchronize the created artifacts. More specifically, the artifacts of cycle N will not match the final service code of cycle N, and they need to be synchronized before they are extended with new service analysis and service design results.

SERVICE OPTIMIZATION PHASE

Here we thoroughly examine the service code in order to improve its performance in the target DPE and thus prepare the telematic service for successful deployment. This phase depends highly on the selected programming language and the target DPE.

The analysis of the proposed methodology up to now clearly reveals that it provides a step-by-step approach from problem definition to the realization of new telecommunications services. For this purpose, the service is studied and described (in a number of service development cycles) at hierarchically related abstraction levels, in the sense that at each level the results achieved at previous levels of abstraction are preserved and refined. Additionally, the use of the object-oriented paradigm is advocated all along the development process. Consequently, this methodology combines the benefits of object-oriented modeling, particularly in terms of scalability and reusability, with those provided by a top-down approach. Moreover, as service requirements are emphasized, such an approach ensures a high level of confidence that users' expectations of the service will be met.

More specifically, the proposed service development methodology has the following benefits:

- It successfully guides service developers during the entire service creation process, offering them a well defined activity roadmap that enables them to proceed in a step-by-step fashion from service requirements to the service code.
- It facilitates the recognition and understanding of typical service engineering problems with respect to a particular service, and assists the deduction and expression of solutions for them in an adequate manner.
- It assists and constrains service developers by taking into account all the peculiarities of telecommunications services and by properly structuring the service creation problem domain.
- It reduces risk, since it increases the likelihood of creating a successful telematic service by emphasizing careful understanding of the service requirements, and promoting appropriate service analysis and service design strategies.
- It creates a detailed process that can be repeated and duplicated by individuals and especially by service development teams.
- It avoids complexity overload and manages the otherwise overwhelming complexity that usually characterizes the development of telematic services, by appropriately modeling services and abstracting to find and examine only the essential details.
- It creates robust and maintainable object-oriented telematic services by adopting fundamental object-oriented principles, and emphasizing service analysis and service design, recognizing that it is cheaper and easier to make changes during these phases than during the service implementation phase.
- It supports reusability at different abstraction levels.

Finally, it must be kept in mind that to obtain the maximum possible productivity gains and exploit the full potential of the methodology, it is not sufficient to apply it in a mechanical manner. On the contrary, an *adaptation* of the methodology to the service developer's attitude, and to the wider organizational mentality and approach regarding telecommunications and information technology in general, is required. In this way, the proposed methodology will be able to support service creation activities even more effectively, without restricting the creativity of service developers, and fully utilizing their prior service development experience.

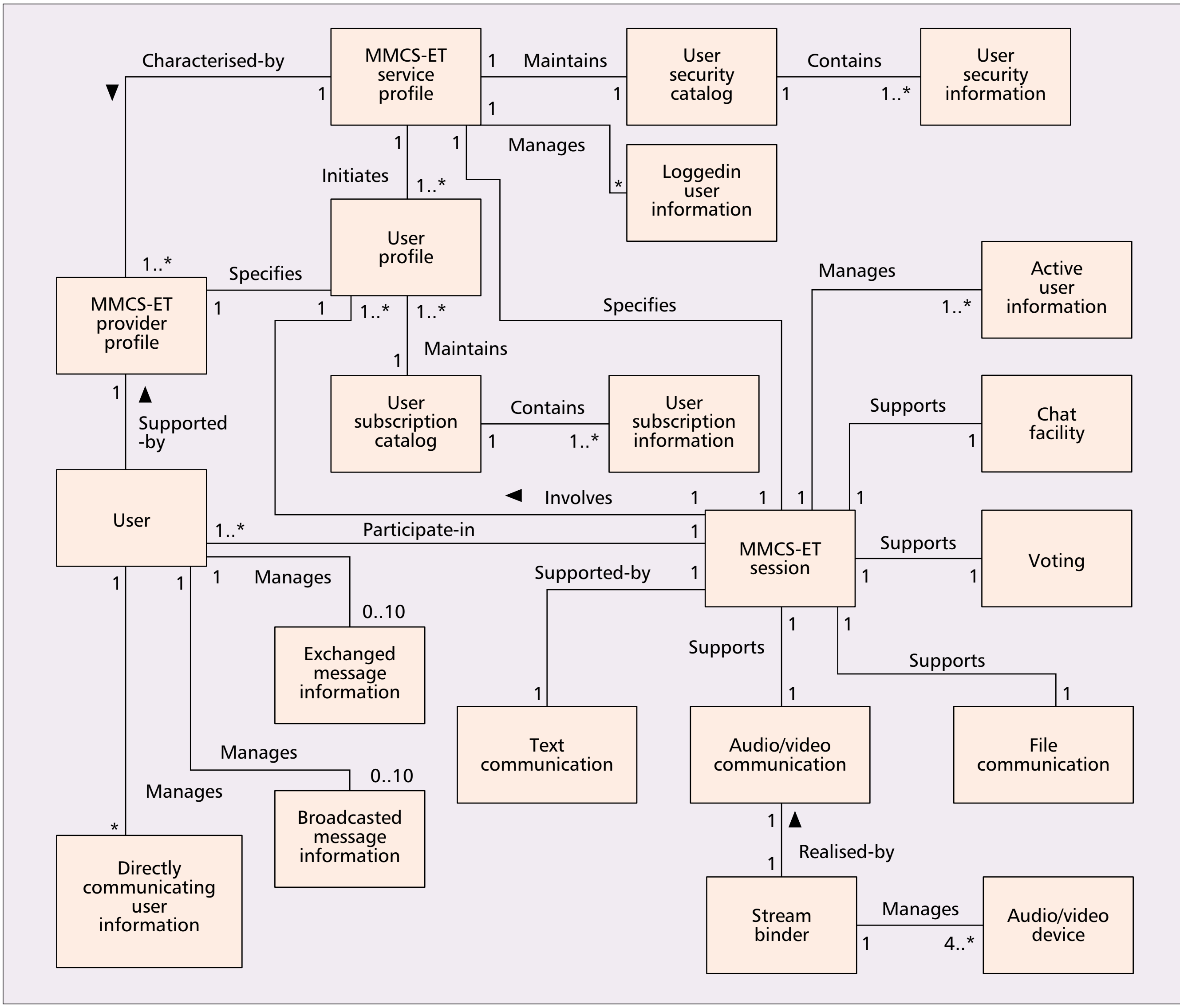
The service optimization phase thoroughly examines the service code in order to improve its performance in the target DPE and thus prepare the telematic service for a successful deployment. This phase depends highly on the selected programming language and the target DPE.

VALIDATION OF THE SERVICE DEVELOPMENT METHODOLOGY

Devising, constructing, and suitably presenting a service development methodology is a complicated task, since it has a wide scope, requires extended skills for the analysis and synthesis of conceptual structures, and involves the consideration of a variety of factors. Furthermore, this task is impossible to view as a pure theoretical exercise, because the success of such a methodology depends on its ability to be applied, in an efficient and effective manner, to the development of actual telematic services, and the only way to take into account and examine this ability is to gain significant *practical experience* on the use of the methodology for various services under different circumstances. For this reason, this section has a more practical nature and attempts to determine the *applicability* and *correctness* of the proposed service development methodology by performing appropriate *validation* activities.

In order to validate the proposed methodology and examine its usefulness, correctness, consistency, flexibility, effectiveness, and efficiency, several simple (but realistic and representative) *service scenarios*, regarding a variety of service creation activities for different simple telecommunications services, were initially considered. The service scenarios considered are distributed collaborative design, distributed case handling, remote monitoring, remote database access, remote database utilization, remote access to expertise, remote application running, entertainment on demand (pay-per-view), remote consultation, and social conversation (chat). Furthermore, in order to verify and reinforce the validation findings under (more) realistic conditions, and also determine the true practical value and applicability of the proposed methodology, all the phases of the methodology were used for the development of a real representative telematic service (a *multimedia conferencing service for education and training, MMCS-ET*).

At the beginning of the first phase of the proposed methodology the *main objective* of the MMCS-ET is specified. This is to enable a teacher/trainer to teach a specific course to a number of geographically dispersed/distributed students/trainees. The service should establish an educational session between the teacher and the remote students equivalent to the educational



■ **Figure 3.** *The main service conceptual model of the MMCS-ET.*

session that would have been established between the same people (teacher and students) in a traditional classroom. Therefore, in a virtual classroom there is a need for audio/video (A/V) communication among all session participants (to substitute for face-to-face contact), text communication between only two session participants (as achieved with notepads), text communication among all session participants (as achieved with a blackboard), file communication between session participants (for exchange of course material), and collaboration among all session participants in order to perform a common task.

By analyzing all the above mentioned (unstructured) requirements a set of *initial service requirements* (categorized into session management, interaction, and collaboration support service requirements), together with a set of service functions are identified. From them the following use cases are deduced:

- Start up the MMCS-ET service
- Log in to the MMCS-ET provider domain
- Start a new MMCS-ET session
- Invite a student to join a MMCS-ET session
- Join a MMCS-ET session after being invited
- Invite a student to direct communication
- Accept direct communication with a student
- Engage in text, file, or A/V communication with an active user
- Start, stop, or terminate A/V communication with an active user
- Engage in a chat file communication with all active users
- Start a voting process between all active users and vote
- Present the outcome of a voting process to all the involved active users
- Terminate direct communication between two active users
- Remove a student from an MMCS-ET session
- Terminate an MMCS-ET session
- Terminate the MMCS-ET service

These use cases are examined in several iterative service development cycles. In the service analysis phase the essential use cases are defined and expressed in an *expanded format*. Based on the expanded use cases, the (*main*) *service con-*

	MMCS-ET code
Number of code lines	7000
Number of COM objects	34
Number of user-defined methods/functions	273
Error checking	Examination of HRESULT
Threading	The simple single threaded model, the single threaded apartment (STA) model, the multithreaded apartment (MTA) model
Reuse of COM objects	Aggregation, containment
Notification of asynchronous events	Callback interfaces, connection points

Table 1. *Implementation characteristics of the MMCS-ET code.*

These validation attempts provided strong evidence that the proposed methodology can be used efficiently for the development of new telecommunications services and that it is effective since it can lead to the desired outcome, that is, a successful telematic service that satisfies the requirements of its users.

ceptual model of Fig. 3 is created for the MMCS-ET, and in an attempt to gain an understanding of the service behavior, a *service sequence diagram* is created for the typical course of events of each of the identified use cases. The effect of the service operations revealed from the service sequence diagrams is described in *service operation contracts*. For each service operation, a service operation contract is constructed.

Taking into account all the artifacts produced so far, in the service design phase a *service interaction diagram* is created for each of the identified service operations. From these service interaction diagrams the way the MMCS-ET service COs communicate via messages in order to fulfill the service requirements is evident. The participating MMCS-ET service objects are drawn from the main service conceptual model of Fig. 3, after taking into account the service components proposed by the TINA-C service architecture, the service requirements regarding the MMCS-ET, and other previously created artifacts. By analyzing the service interaction diagrams of the MMCS-ET, all the service classes (together with their attributes and methods) participating in the software realization of the MMCS-ET are identified and illustrated in a *service design class diagram*.

Considering all the artifacts produced in the service design phase, the MMCS-ET was implemented on MS Windows NT 4.0 using Microsoft's Visual C++ (v. 6.0) together with Microsoft's DCOM [8], appropriately extended with a high-level application programming interface (API) in order to support continuous media interactions, and was executed on a number of workstations connected via a 10 Mb/s Ethernet LAN [4]. All the interconnected workstations belong to the same (MS Windows NT) domain; one of them functions as a primary domain controller.

The service code produced after implementing all the service classes of the service design class diagram of the MMCS-ET has the characteristics of Table 1. The first three characteristics (number of code lines, number of COM objects, and number of user-defined methods/functions) reveal the complexity and difficulty of implementing the MMCS-ET, especially considering that at the beginning of this validation attempt DCOM was a very new technology. The other implementation characteristics included in Table 1 are related to the exploitation of certain DCOM capabilities in the MMCS-ET code.

These validation attempts revealed and provided strong evidence that the proposed methodology can be used efficiently for the development of new telecommunications services and that it is correct and effective since it can lead to the desired outcome, that is, a successful telematic service that satisfies the requirements of its users. Additionally, they provided valuable *feedback*, enlightened many aspects regarding the proposed process (in terms of phases, activities, and steps) and the modeling approach followed, revealed different perspectives for considering some matters, pointed out a number of deficiencies, and in general resulted in further improvement of the methodology.

Summary and Conclusions

The telecommunications industry is currently facing a number of challenges imposed by changes in the telecommunications market. Deregulation, liberalization, and competition, fueled by rapid technological developments, imply requirements for higher utilization of the network infrastructure, shorter time to market for new telecommunications services, a much higher degree of customization of these services, cost reduction of service development, open network provision, global connectivity, and global information access. Furthermore, both telecommunications services and networks are ever growing in sophistication and complexity with a tendency to become large-scale highly decentralized and heterogeneous systems involving numerous users and resources. All these changes require more complex software systems and thus make evident the need to accelerate the integration of information technology and telecommunications.

The proposed telecommunications service engineering framework promises to facilitate such an integration, strengthens the belief that telecommunication networks and data networks are converging, and reveals that this convergence is also manifested in the service engineering realm. With the use of the proposed framework, paradigms of software development and associated practices can more easily be diffused between telecommunications and data networks. The proposed framework facilitates this transition from the legacy telecommunications world since it proposes the use of distributed object platforms as "signaling" infrastructures for the realization of unified service control and management. In this way, it enables the provision of

The proposed framework facilitates this transition from the legacy telecommunications world since it proposes the use of distributed object platforms as "signaling" infrastructures for the realization of unified service control and management.

a great variety of new sophisticated *telematic services*, that will eventually realize the open services market, creating a ubiquitous information and communication system available to, and usable by, all.

References

[1] H. Akimaru, M. R. Finley, and Z. Niu, "Elements of the Emerging Broadband Information Highway," *IEEE Commun. Mag.*, vol. 35, no. 6, June 1997, pp. 84–94.

[2] K. G. August, V. B. Lawrence, and B. R. Saltzberg, "An Introduction to Future Communications Services and Access," *Bell Labs Tech. J.*, April–June 1999, pp. 3–20.

[3] R. Saracco, "A Vision of Software Evolution in Telecommunications Networks," *CSELT Tech. Rep.*, vol. XXIV, no. 5, Oct. 1996, pp. 829–47.

[4] D. X. Adamopoulos, G. Pavlou, and C. A. Papandreou, "Supporting Advanced Multimedia Telecommunications Services Using the Distributed Component Object Model," *Proc. 7th Int'l. Conf. Intelligence in Services and Networks*, Lecture Notes in Comp. Science, vol. 1774, Springer-Verlag, Berlin, 2000, pp. 89–104.

[5] H. Berndt, T. Hamada, and P. Graubmann, "TINA: Its Achievements and its Future Directions," *IEEE Commun. Surveys and Tutorials*, vol. 3, no. 1, 1st qtr., 2000.

[6] I. Venieris, F. Zizza, and T. Magedanz, Eds., *Object Oriented Software Technologies in Telecommunications: From Theory to Practice*, Wiley, 2000.

[7] P. Evits, "A UML Pattern Language," Macmillan Technology Series, Feb. 2000.

[8] N. Brown and C. Kindel, "Distributed Component Object Model Protocol – DCOM," Microsoft Corp., Jan. 1998.

Biographies

DIONISIS X. ADAMOPOULOS (D.Adamopoulos@surrey.ac.uk) holds a diploma in computer science (1993) from the Athens University of Economics and Business, Greece, an M.Sc. degree with distinction (1994), and a Ph.D. degree (2000), both in teleinformatics (telematics), from the Department of Electronic Engineering of the University of Surrey, England. He is currently a collaborative researcher at the Center for Communication Systems Research (CCSR) of the University of Surrey. His research interests include service engineering, distributed multimedia, object-oriented analysis and design, distributed object platforms, pervasive computing, and telecommunications management.

GEORGE PAVLOU (G.Pavlou@surrey.ac.uk) gained a diploma in electrical engineering from the National Technical University of Athens. He obtained M.Sc. and Ph.D. degrees in computer science from University College London, where he acted as a senior research fellow and lecturer. He is currently professor of information networking in the CCSR at the University of Surrey, where he leads the activities of the networks research group. Over the last 10 years he has led a number of collaborative research projects in networking, with emphasis on management and control. Routing and quality of service are among his key research areas of interest. He has contributed to ISO and ITU-T standardization work.

CONSTANTINE A. PAPANDREOU (kospap@ote.gr) holds a Dipl. Ing. degree as well as a postgraduate degree in engineering economics, both from the Technical University of Munich. He holds also a Ph.D. degree in telematics from the University of Munich. He has served since 1970 in the Hellenic Telecommunications Organization (OTE) as an expert in telecommunications and informatics in various positions. Since 1992 he has been a director in OTE. In parallel to these activities he has been teaching teleinformatics and information technology for several years at the Athens University of Economics and Business, the University of Piraeus, and the Higher School of Telecommunications of OTE. Service engineering, information systems management, telecommunications policy, education and training, multimedia communications, and office automation are among his key research areas of interest.