Towards the Efficient Support of Telecommunications Service Engineering Activities

Dionisis X. Adamopoulos, George Pavlou
Centre for Communication Systems Research (CCSR)
University of Surrey, Guildford, GU2 7XH
{D.Adamopoulos, G.Pavlou}@eim.surrey.ac.uk

Constantine A. Papandreou
Hellenic Telecommunications Organisation (OTE)
17 Kalliga Street, GR-114 73, Athens, Greece
kospap@org.ote.gr

Abstract: Continued global deregulation and the harnessing of new technologies has had an enormous impact on the telecommunications industry, leading to a dramatic increase in the number and type of services that telecommunication companies can offer. Building new advanced multimedia telecommunications services in a distributed and heterogeneous environment is very difficult, unless there is a methodology to support the entire service development process in a structured and systematic manner, and assist and constrain service designers and developers by setting out goals and providing specific means to achieve these goals. Therefore, this paper, responding to the requirements of the future open telecommunications service market, proposes a service creation methodology and presents it by emphasising on a number of related important methodological issues. The application of this methodology to the development of a multimedia conferencing service for education and training is then examined.

1. Introduction.

The advent of deregulation combined with new opportunities opened by advances in telecommunications technologies has dramatically changed the paradigm of telecommunications services. Due to the evolving synergy between information and telecommunications technology, termed telematics, that is gradually gaining momentum, the number and capabilities of services is rapidly growing, and services and network infrastructure is expected to be more and more unbundled. Services have become increasingly software based, taking the form of distributed (multimedia) applications, operating on Distributed Processing Environments (DPEs), and reusing and combining other already deployed services as well as capabilities of telecommunication and computing resources [1][2][4].

The timely availability of these new telecommunications services (telematic services) corresponding to the needs of the market is an important condition for gaining a competitive advantage and for the exploitation of the huge telecommunication potential that will be offered by the forthcoming integrated (fixed and mobile) broadband networks. Thus, the main strengths of a service provider is the diversity of its service portfolio and the ability to respond to new market demands quickly. In such a competitive environment efficient service creation enables the fast launching of new services without compromising the overall service quality [6][8].

The traditional waterfall model for software development with its sequential phases of requirements capture, design, implementation and testing is inadequate to support the development of telecommunications services. Its limitations lie in the lack of support for specific telecommunication features and such advanced concepts as reuse, the exploitation of rapid prototyping techniques for iterative requirements specification, and the evolutionary nature of service creation. Thus, a process model in the form of a methodology, taking into account these points, and covering the complete life-cycle of service creation will be proposed and examined in this paper.

2. The Proposed Service Development Methodology.

In a telecommunications world currently undergoing significant regulatory changes and permanently facing technical innovations, telecommunications operators need to master the complexity of service software, in order to satisfy the highly diversified market demands and acquire a competitive advantage by quickly and economically developing and introducing a broad range of new services [5]. To achieve such an ambitious, yet strategic to the telecommunications operators goal, a service creation methodology based on the rich conceptual model of the open service architectural framework specified by the Telecommunications Information Networking Architecture Consortium (TINA-C) [9], is proposed.

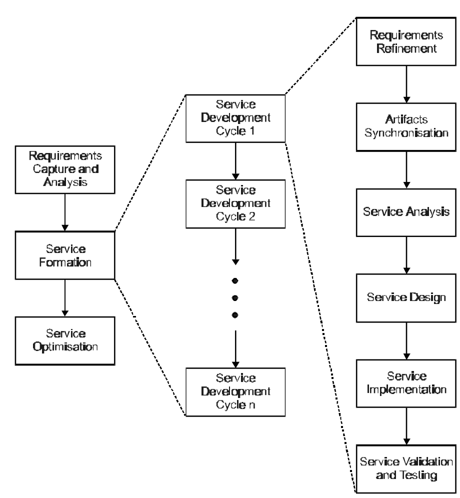

Fig. 1: Iterative service development cycles in the proposed methodology.

A high-level or macro-level view of the proposed service creation methodology can be seen in Fig. 1. 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. The proposed use of the Unified Modelling Language (UML) notation [3] 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.

According to Fig. 1 the main phases of the proposed methodology are the following:

- *Requirements capture and analysis phase:* It identifies the telematic service requirements (together with a number of roles), and presents them in a structured way. The use case concept is applied for this purpose.
- *Service analysis phase:* It describes the semantics of the problem domain that the telematic service is designed for. Thus, it identifies the objects that compose a service (information service objects), their types, and their relationships. This phase uses UML static structure diagrams, UML sequence diagrams (accompanied with the definition of pre- and post- conditions of service operations), and UML statechart diagrams.
- *Service design phase:* It produces the design specifications of the telematic service under examination. Computational modelling is taking place in this phase and thus the service is described in terms of TINA-C computational objects interacting with each other. The notation chosen is the TINA-C Object Definition Language (ODL), together with UML collaboration, sequence, and class diagrams.
- *Service implementation phase:* In this phase the pieces of the service software (Computational Objects, COs) are defined and implemented in an object-oriented programming language (e.g. C++, Java), inside a TINA-C compliant DPE. This phase uses UML packages and UML component and deployment diagrams.
- *Service validation and testing phase:* It subjects the implemented telematic service in a variety of tests to ensure its correct and reliable operation.
- *Service optimisation phase:* It examines thoroughly the service code in order to improve its performance in the target DPE, and thus prepare the telematic service for a successful deployment. In this phase the UML component and deployment diagrams developed in the service implementation phase are refined.

As can be seen from Fig. 1, the proposed methodology is conceptually consistent with the viewpoint separation as advocated by TINA-C in accordance with the Reference Model for Open Distributed Processing (RM-ODP). 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, a number of important issues regarding the proposed methodology are considered and examined, together with an attempt to apply all the phases of the methodology to the development of a specific characteristic telematic service.

2.1. Important Methodological Considerations.

The proposed methodology specifies a service development process from the identification of service requirements through to the actual implementation of a telematic service. Its most important phases are the requirements capture and analysis phase, and the phases of service analysis and service design which are performed inside repeated service development cycles. A useful approach is to bound each of these cycles within a rigidly fixed time frame (a time-box). All work related to a specific cycle must be accomplished in that time frame.

To succeed with a time-box schedule, it is necessary to choose the service requirements carefully, as iterative service development cycles are organised according to the identified use case requirements. More specifically, a service development cycle is assigned to implement one or more use cases, or simplified versions of use cases (which is quite common when the complete use case is too complex to tackle in one cycle). Use cases should be ranked and high ranking use cases should be tackled in early service development cycles. Furthermore, use cases that (is expected to) significantly influence the core service architecture, or which are critical and / or high-risk, should be considered first.

Service development is complex and therefore decomposition ("divide-and-conquer") is the primary strategy considered to deal with this complexity by breaking a telematic service up into manageable units. The proposed methodology applies object-oriented analysis and design. Thus, during the (object-oriented) service analysis phase, there is an emphasis on finding and describing the service concepts (service information objects) in the service domain, whose boundaries are determined, as accurately as possible, by the telematic service under examination during the requirements capture and analysis phase. Furthermore, during the (object-oriented) service design phase, there is an emphasis on defining logical (software) service objects (service COs or service components) with attributes and methods that will ultimately be implemented in an object-oriented programming language, such as C++ or Java, inside a TINA-C compliant DPE during the service implementation phase.

It has to be stressed that the division between service analysis and service design is fuzzy. In telecommunications service engineering (and sometimes in other disciplines also), analysis and design work exists on a continuum, and different practitioners of service analysis and service design classify an activity at varying points on the continuum [5][8]. Therefore, it is not helpful being rigid about what constitutes a service analysis versus a service design step.

Irrespective of the exact scope of service analysis and service design, the most important ability in both these two phases is to successfully assign responsibilities to service components [4][5]. This is the most critical skill, because this activity must be performed (it is inescapable) and it has the most profound effect on the robustness, maintainability, and reusability of the resulting (implemented in software) service components, which are the main building blocks of telematic services. Assigning responsibilities is inevitable even when a service developer hasn't got the opportunity to perform any other service analysis or service design activities (a "rush to code" service development process).

Finally, the timing of the creation of some artifacts during the application of the proposed methodology needs to be discussed and examined. More specifically, certain artifacts created during the service analysis phase, such as a service conceptual model and expanded use cases, may also be created during the requirements capture and analysis phase. In both cases there are trade-offs in terms of the benefit of the early acquisition of information versus facing too much complexity.

2.2. Application of the Methodology.

In order to validate the proposed service development methodology and examine its usefulness, correctness, consistency, flexibility, and efficiency, several simple scenarios, regarding a variety of service creation activities for different simple telecommunications services, were considered. These scenarios confirmed that the methodology has all the above anticipated positive characteristics.

To verify and reinforce these findings under (more) realistic conditions, and to determine also the true practical value and applicability of the proposed methodology, all the phases of the methodology were used for the development of a real telematic service (a MultiMedia Conferencing Service for Education and Training, MMCS-ET) that is expected to have great demand in the near future [7]. This validation attempt, which provided valuable feedback and resulted to the further improvement of the methodology, is described briefly in this section, focusing mainly on the most important artifacts that were created during the application of the methodology. The intention is to provide characteristic examples on the use of the methodology to those interested in employing it for the development of new telecommunications services (e.g. service developers, service designers, etc.) and increase in that way their understanding of the methodology and their confidence on its effectiveness.

During the first phase of the methodology it was found that the main requirement was to develop a telematic service (MMCS-ET) that will enable a teacher / trainer to teach efficiently and effectively a specific course to a number of geographically dispersed / distributed students / trainees. The service should establish an educational / training session between the teacher and the remote students that is equivalent to the educational / training session that would have been established between the same people (teacher and students) in a traditional classroom.

By analysing all the collected (unstructured) requirements a set of initial service requirements (categorised into session management, interaction, and collaboration support service requirements), together with a set of service functions were identified. From them a number of use cases were deduced.

These uses cases were considered in several iterative service development cycles. However, in this section, for reasons of clarity and simplicity, only one characteristic use case ("Invite a student to join a MMCS-ET session") is examined (target use case) in the same way as if it was only one service development cycle.

All use cases were expressed in an expanded format. The middle section of the expanded use cases (the "Typical Course of Events") is the most important as it describes in detail the required interaction between the actors and the telematic service. A critical aspect of this section is that it describes the most common sequence of activities needed for the successful completion of a service process. Alternative situations or exceptions that may arise with respect to the typical course were included in the final section of the expanded use cases (the "Alternative Course of Events").

Based on the expanded use cases, in the service analysis phase, the service conceptual model of Fig. 2 was created for the MMCS-ET. More specifically, the noun phrases in the text of the expanded use cases were considered as candidate service concepts and attributes. Furthermore, as an attribute is a logical data value of a service object, the service conceptual model included all the attributes for which the service requirements suggested or implied a need to remember information. Service concepts are related by associations which indicate some meaningful and interesting connection. Therefore, the service conceptual model of the MMCS-ET included all the associations for which knowledge of the corresponding relationship needs to be preserved for some duration ("need-to-know" associations). It has to be noted that it is generally undesirable to overwhelm the service conceptual model with associations that are not strongly required and which do not increase understanding.

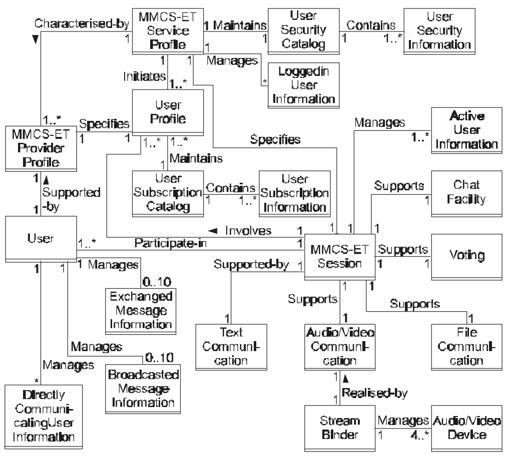

Fig. 2: The service conceptual model of the MMCS-ET.

Use cases suggest how actors interact with the telematic service under examination. During this interaction an actor generates events to the telematic service, requesting some operation in response. It is desirable to isolate and illustrate the operations that an actor requests from a telematic service (service operations) in service sequence diagrams, because they are an important part of understanding service behaviour.

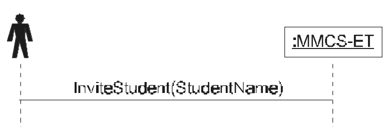

Fig. 3: A service sequence diagram for the MMCS-ET.

Therefore, a service sequence diagram (see Fig. 3) was created for the typical course of events of each one of the identified use cases in the following way:

- A vertical line was drawn representing the MMCS-ET as a black box.
- Each actor that directly operated on the MMCS-ET was identified and a vertical line was drawn for him / her.
- From the use case typical course of events text, the (external) service events that each actor generates were identified and illustrated in the correct order on the diagram.

The effect of the service operations that were revealed from the service sequence diagrams was described in service operation contracts. For each service operation, a service operation contract was constructed. Its "Responsibilities" section informally describes the purpose of the service operation, while its "Post-conditions" section declaratively describes the state changes that occur to service objects in the service conceptual model of Fig. 2, using a number of suitably selected statements (instance creation, instance deletion, attribute modification, association formed, association broken, and user interface activation).

Taking into account all the artifacts produced so far, in the service design phase, a service interaction diagram in the form of a UML collaboration diagram was created for each one of the identified service operations. The objective was to fulfill the post-conditions of the corresponding service operation contracts, recognising however that the previously defined post-conditions are merely an initial best guess or estimate of what must be achieved, and therefore their accuracy should be questioned.

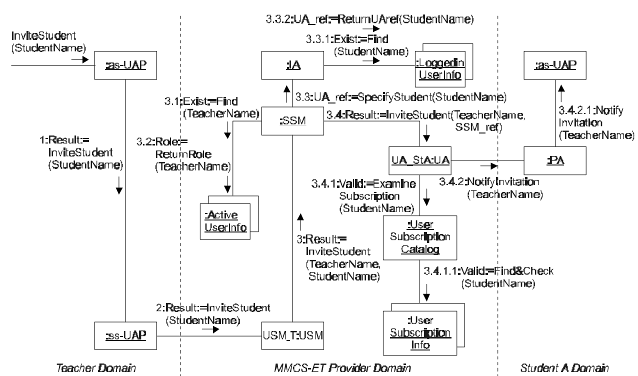

Fig. 4: Service interaction diagram for inviting a student to join a MMCS-ET session.

From these service interaction diagrams (an example of which is depicted in Fig. 4 for the target use case) the way that the MMCS-ET service (computational) objects communicate via messages in order to fulfill the service requirements is evident. The participating MMCS-ET service objects were drawn from the service conceptual model of Fig. 2, after taking into account the service components proposed by the TINA-C service architecture [9]. Therefore, the links between the MMCS-ET service objects are actually instances of the associations present in the service conceptual model of Fig. 2, represent connection paths between service object instances, and indicate that some form of navigation and visibility between the instances is possible (attribute, parameter, locally declared or global visibility).

By analysing the service interaction diagrams, all the service classes (together with their attributes and methods) participating in the software realisation of the MMCS-ET were identified and illustrated (with simplifications) in the service design class diagram of Fig. 5. The associations present in this diagram satisfy the ongoing "memory needs" revealed by the service interaction diagrams and the navigability arrows on them indicate the direction of attribute visibility (non-attribute visibility is indicated by dependency relationships).

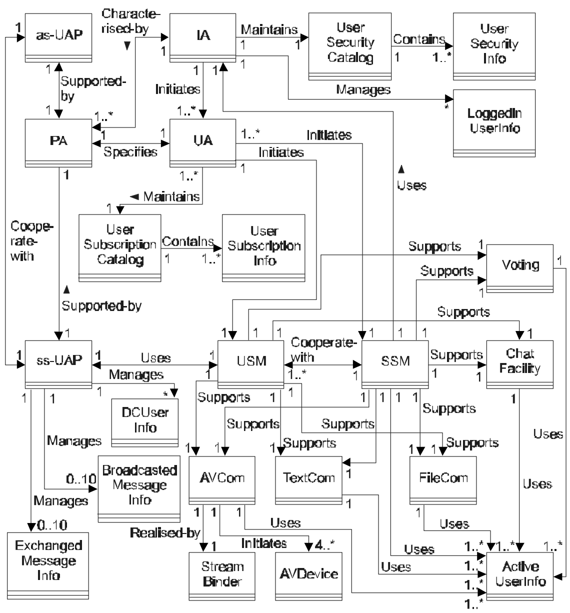

Fig. 5: The service design class diagram of the MMCS-ET (simplified).

Considering all the artifacts produced in the service design phase, the MMCS-ET was implemented using Microsoft's Visual C++ together with Microsoft's Distributed Component Object Model (DCOM) (appropriately extended with a high-level API in order to support continuous media interactions) on MS Windows NT 4.0, and was executed on a number of workstations connected via a 10 Mbit/s Ethernet LAN [2][7]. All the interconnected workstations belong to the same (MS Windows NT) domain and one of them functions as a primary domain controller.

3. Conclusions.

A revolution in information technology and telecommunications is already in progress, and is expected to escalate rapidly in the near future. The two worlds have already begun to converge and the convergence path is marked by the continuously expanding penetration and scope of telecommunications services. Considering this evolution, emerging telecommunications systems promise to offer a wide variety of highly sophisticated, personalised, and ubiquitous services over the widest possible coverage area. Several providers are involved in such an ambitious (yet realistic) service provision scenario in which competition will mostly focus at the service level, with multiple providers offering new services to the market in a short time over a variety of networks and end systems. In the light of these challenges and because of the highly increasing complexity of new telecommunications services and the inherent distributed nature of them, a methodology covering the whole service development process, like the one proposed and examined in this paper, is absolutely necessary.

The application of the proposed service creation methodology to the development of the MMCS-ET has enlightened many aspects regarding its structure and its use, and offered confidence that it can enable the fast and efficient creation of telematic services. It must be kept in mind, however, that to obtain the maximum possible productivity gains and to 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 organisational mentality and approach regarding telecommunications, is required. In that way, the proposed methodology will be able to support service creation activities even more effectively, without restricting the creativity of service developers, and by utilising fully their prior service development experience.

References

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

[2] Adamopoulos, D.X., Pavlou, G., Papandreou, C.A., "Supporting Advanced Multimedia Telecommunications Services Using the Distributed Component Object Model", Proceedings of IS&N 2000, LNCS, Vol. 1774, Springer-Verlag, 2000, pp. 89-104.

[3] Booch, G., Rumbaugh, J., Jacobson, I., "Unified Modelling Language User Guide", ACM Press, 1998.

[4] De la Fuente, L.A., Ferrari, L., Gallego, J., Llamas, P., "The Eurescom P610 Project: Providing Framework, Architecture and Methodology for Multimedia Services Management", Proceedings of IFIP/IEEE DSOM '97, Sydney, Australia, 1997, pp. 145-154.

[5] Declan, M., "Adopting Object Oriented Analysis for Telecommunications Systems Development", Proceedings of IS&N '97, LNCS, Vol. 1238, Springer-Verlag, 1997, pp. 117-125.

[6] Demestichas, P., et al., "Issues in Service Creation for Open Distributed Processing Environments", Proceedings of ICC '99, June 1999.

[7] Papandreou, C.A., Adamopoulos, D.X., "Design of an Interactive Teletraining System", BT Engineering, Vol. 17, 1998, pp. 175-181.

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

[9] TINA-C, "Definition of Service Architecture", Version 5.0, 1997.