Exploiting Distributed Object Technology for the Design of Net-Centric Services

Dr. DIONISIS X. ADAMOPOULOS
University of Piraeus, GREECE and
Centre for Communication Systems Research (CCSR)
University of Surrey, ENGLAND

Dr. CONSTANTINE A. PAPANDREOU
University of Piraeus
GREECE

Abstract: - In the highly competitive telecommunications environment of today a service provider must continually evolve its network and enable new revenue-generating services faster and more cost effectively than the competition. These services should create value for the end users by satisfying all their functional requirements in an efficient manner, and they should be delivered with simplicity and ease of provisioning. This paper focuses on the transition from a requirements capture and analysis phase to a service analysis phase in the framework of an object-oriented service creation methodology, by considering the internal structure and the functionality of a telematic service. After the examination of the main activities that take place in the service analysis phase and the identification of the main artifacts that are produced during them, the overall service development process is highlighted, and the paper attempts to validate, through several service creation experiments, not only the service analysis phase, but also the overall service creation methodology. Finally, the validation results are discussed and further evaluation actions are briefly outlined.

Key-Words: - Service engineering, service creation, distributed object computing

1 Introduction

Increased competition, regulatory changes and the convergence of network technologies are causing service providers to look for innovative telecommunications services (telematic services) to differentiate their offerings and gain a competitive advantage. Telecommunication networks are gradually evolving towards integrated services networks or Multi-Service Networks (MSNs), which are networks capable of supporting a wide range of services. In MSNs, services are viewed as value adding distributed applications, composed using more elementary facilities available underneath through specialised Application Programming Interfaces (APIs) and operating on top of a general purpose communications subsystem.

For this reason, the emphasis is placed on the telecommunication software and on the rapid development of services upon open, programmable networks. Therefore, this paper proposes a structured service creation approach, emphasizing service analysis activities, that offers a viable service paradigm inside an open deregulated multi-provider telecommunications market place and is compatible with and influenced by the state of the art service creation technologies of Open Service Access (OSA), Parlay and Java APIs for Integrated Networks (JAIN) [6], and conformant to the open service architectural framework specified by the Telecommunications Information Networking Architecture Consortium (TINA-C) [2][8]. The practical usefulness and efficiency of the proposed approach is ensured by extensive validation attempts with various services, involving comparative examination and experimentation with the use of different development environments.

2 Analysing Service Requirements

Service analysis activities constitute the service analysis phase in the service creation methodology, which is proposed to have an iterative and incremental, use case driven character. 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 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 service analysis, service design, service implementation and validation, and service testing. The service grows incrementally as each cycle is completed.

The aim of the service analysis phase is to determine the functionality needed for satisfying the service requirements 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 [1].

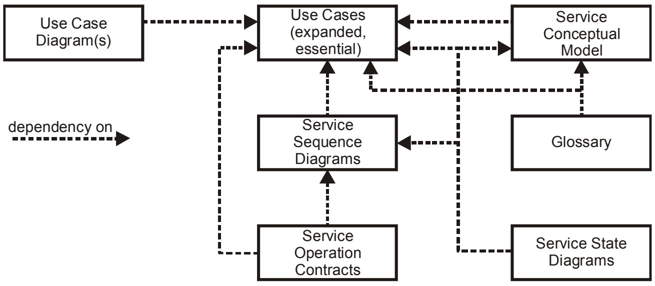

Fig. 1: Service analysis phase artifact dependencies.

The most important activities of this phase are examined in the following sections. The dependencies between the artifacts produced can be seen in Fig. 1.

A. Definition of Service Conceptual Models

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. The resulting (main) service conceptual model involves identifying a rich set of service concepts (Information Objects, IOs) regarding the service under examination by investigating the service domain and by analysing the essential use cases [7].

The main service conceptual model is accompanied by a set of ancillary service conceptual models. These models are derived by (and correspond to) a number of generic information models deduced from the TINA-C service architecture [8] and complement semantically the main service conceptual model with useful session related concepts and structures [3].

The following steps specify the main service conceptual model:

Step 1: Identify the service concepts.

A central task when creating a service conceptual model is the identification of the service concepts. Two techniques are proposed for the identification of service concepts. The first is based on the use of a service concept category list, which contains categories that are usually worth considering, though not in any particular order of importance. Another useful technique is to consider the noun phrases in the text of the expanded use cases as candidate service concepts or attributes.

Step 2: Identify associations between the service concepts.

After identifying the service concepts, it is also necessary to identify those associations of the service concepts that are needed to satisfy the information requirements of the current use case(s) under development and which aid the comprehension of the service conceptual model. The associations that should be considered in order to be included in a service conceptual model are the associations for which the service requirements suggest or imply that knowledge of the relationship that they present needs to be preserved for some duration ("need-to-know" associations) or are otherwise strongly suggested in the service developer's perception of the problem domain.

Step 3: Identify attributes of the service concepts.

A service conceptual model should include all the attributes of the identified service concepts for which the service requirements suggest or imply a need to remember information. These attributes should preferably be simple attributes or pure data values. Caution is needed to avoid modelling a (complex) service concept as an attribute or relating two service concepts with an attribute instead of an association.

Step 4: Draw the main service conceptual model.

Adding the identified type hierarchies, associations and attributes to the initial service conceptual model, forms the main service conceptual model. It has to be noted that a verb phrase should be used for naming an association, in such a way that the association's name together with the names of the service concepts that it relates create a sequence that is readable and meaningful.

The proposed methodology considers the TINA-C service architecture (which has a direct and significant influence to subsequent service creation technologies) in a critical manner with the intention to extract from it useful concepts and guidelines / techniques. Taking into account the generic TINA-C session related information models [1] and the different types of sessions that can be established between business administrative domains, access sessions can be classified according to the specialisation hierarchy shown in Fig. 2(a).

The access session related service IOs and their relationships are depicted in the information model of Fig. 2(b). In this figure, the Domain Access Session (D_AS) service IO is associated with a particular domain and represents the generic information required to establish and support access interactions between two domains. Furthermore, it is specialised into UD_AS (managed by the user), PD_AS (managed by the provider) and PeerD_AS service IOs, as each D_AS is associated with a particular access role. All information that is used directly by the D_AS for authorisation decisions, constraints and customisation of the D_ASs, Access Sessions and Service Sessions is contained in the User Profile.

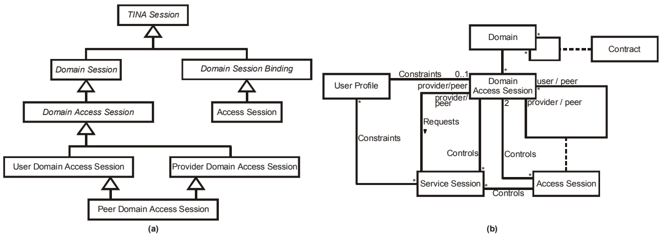

Fig. 2: Important access session related information models:
(a) Classification of the access session,
(b) The access session information model.

Service sessions can be classified according to the specialisation hierarchy shown in Figure 3(a). The service session related service IOs and their relationships are depicted in the information model of Figure 3(b). Every service session consists of usage and provider service sessions. Each member of a session, i.e. an end-user, a resource or another session, is associated with a usage service session. Furthermore, each usage service session can extend over two domains and is composed of two complementary Domain Usage Service Sessions (D_USSs). The Domain Usage Service Session Binding (D_USS Binding) represents the dynamic information associated with the binding of two D_USSs.

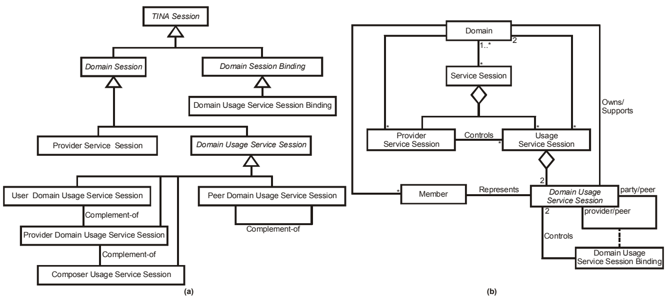

Fig. 3: Important service session related information models:
(a) Classification of the service session,
(b) The service session information model.

B. Definition of Important Artifacts

Before proceeding to a logical design of how a telematic service will work in terms of software components, its behaviour is necessary to be examined and defined as a black box. In this way, service behaviour is considered as a description of what the telematic service does, without explaining how it does it. One part of that description is service sequence diagrams.

A service sequence diagram should be done for the typical course of events of each use case and sometimes for the most important alternative courses. It depicts, for a particular course of events within a use case, the external actors that interact directly with the service, the service (as a black box), and the service events that the actors generate. Service events (and their associated service operations) should be expressed in an abstract way, emphasising their intention, and not in an implementation specific manner [7].

The behaviour of a service is further defined by service operation contracts, as they describe the effect of service operations upon the service. Their creation is dependent on the prior development of use cases and service sequence diagrams, and on the identification of service operations. UML contains support for defining service contracts by allowing the definition of pre- and post-conditions of service operations [4].

Service state diagrams can successfully describe the legal sequence of external service events that are recognised and handled by a telematic service in the context of a specific use case. These are UML state diagrams, which illustrate the interesting and significant service events and the states of a telematic service, together with the behaviour of the service in reaction to a particular service event. A service state diagram which depicts the (overall) service events and their desired sequence within a use case is called a use case service state diagram, and can be created for a specific use case at varying levels of detail depending on the exact modelling needs.

The real value of use case service state diagrams is appreciated when they model complex use cases with many service events, because then they help considerably the service developer(s) during the service design to avoid out-of-sequence service events and the corresponding error conditions. However, use case service state diagrams are not necessary if there is no significant service event ordering. Therefore, their definition in the service analysis phase is optional. In such cases, another (optional again) alternative is the creation of a global service state diagram, which illustrates, for the entire telematic service, all the transitions for service events across all the use cases. It is a union of all the use case service state diagrams and is useful as long as the total number of service events is small enough to keep the diagram comprehensible.

3 The Validation Approach

The proposed service creation methodology (and thus its service analysis phase) was validated and its true practical value and applicability was ensured as it was applied to the design and development of a real complex representative telematic service (a MultiMedia Conferencing Service for Education and Training, MMCS-ET). More specifically, a variety of scenarios / use cases were considered involving the support of session management requirements (session establishment, modification, suspension, resumption, and shutdown), interaction requirements (audio / video, text, and file communication), and collaboration support requirements (chat facility, file exchange facility, and voting), as can be seen in Fig. 4. Due to the incremental and iterative nature of the proposed methodology these use cases were examined in nine (9) service development cycles covering a time period of almost two (2) years.

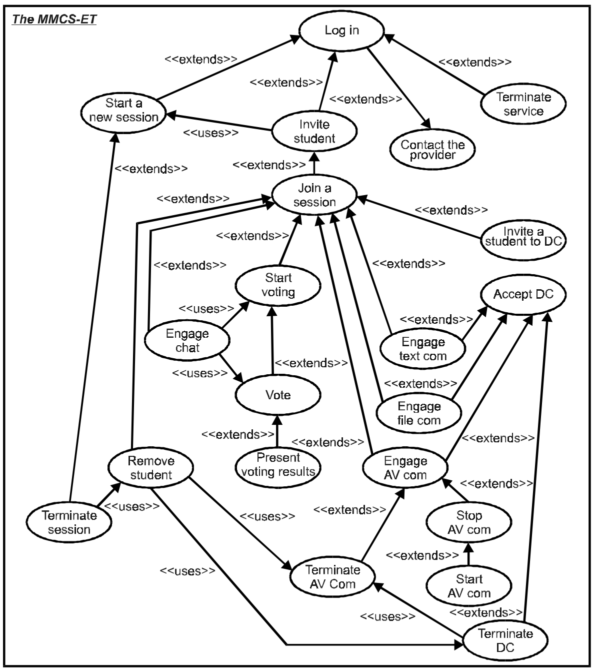

Fig. 4: The simplified use case diagram of the MMCS-ET.

Further validation attempts of the proposed methodology and examination of its usefulness, correctness, consistency, flexibility, effectiveness and efficiency, involved a variety of service creation activities for different telecommunications services (actually service scenarios) using different development approaches. More specifically, the following telematic services were considered: Distributed collaborative design, distributed case handling, remote monitoring, remote database access, remote database utilisation, remote access to expertise, remote application running, entertainment on demand (pay-per-view), remote consultation, social conversation.

These service scenarios were developed in a small scale (with a maximum of five end-users) in both a business and an academic environment, by three (different) teams of two service developers in three different (object-oriented) manners; namely using the proposed service creation methodology, using an ad hoc approach and using a widely accepted general purpose object-oriented software development methodology (the Unified Process, UP). All service developers had similar knowledge and experience, and a manager monitored the service requirements and the reuse of the specifications and code in all cases in order to keep them at comparable levels during the application of the different service creation approaches for each service scenario.

The parameters that were examined each time a telematic service was developed (some of which are explicitly related to service analysis activities) are the following:

- The total time needed for the development of the service from the beginning of the service project until service deployment.
- The number of problems that were reported by the users of the service (e.g. not supported functions, unsatisfactory operation, unpredictable behaviour, etc.) after using it in a daily basis for two months after its deployment.
- The number of service concepts in the service conceptual models, the number of essential use cases, the number of service sequence diagrams, the number of service operation contracts, and the number of service state diagrams in which changes were performed during the first two months of service operation.
- The number of objects in the service code that was necessary to change during the first two months of service operation.
- The number of lines of service code that were added during the first two months of service operation.
- The total time needed for service maintenance activities during the first two months of service operation.
- The total time that the service was necessary to be inactive due to service maintenance during the first two months of service operation.

From the results of the service creation experiments is evident that the ad hoc approach, although it seems to be fast in some cases, is the less flexible approach with the highest risk in misinterpreting the requirements and the highest possibility to cause maintenance problems. The most important drawback of the UP is its difficulty to be applied in a service creation context, which is reflected by the greater total development time, by the more changes that are needed to service analysis artifacts when performing corrections or extensions and by the relatively verbose service code that it produces. Therefore, the proposed process is the most flexible and cost effective approach (in terms of total development time and maintenance problems), satisfies the requirements of the users (slightly better than UP), and (when considering all the parameters) improves the productivity of the service developers and increases the possibility for the successful completion of a service creation project.

5 Concluding Remarks

The validation attempts described in Section 3 provided strong evidence that the proposed methodology and the corresponding service analysis activities can be used efficiently for the development of new telecommunications services in open, programmable service-driven next-generation networks and that they are correct and effective as they can lead to the desired outcome, i.e. a successful telematic service that satisfies the requirements of its users. However, the proposed methodology, apart from being specialised for service creation purposes in the area of telecommunications service engineering, remains a methodology for the development of systems. Therefore, its evaluation using the Normative Information Model-based Systems Analysis and Design (NIMSAD) framework is suggested [5]. Such an attempt will provide additional confidence on the capabilities and the quality of the proposed service creation process and will support the increased expectations regarding its value and impact.

References:

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

[2] Berndt, H., Hamada, T., Graubmann, P., "TINA: Its Achievements and its Future Directions", *IEEE Communications Surveys & Tutorials*, Vol. 1, 2000.

[3] Choi, S., Turner, J. Wolf, T., "Configuring Sessions in Programmable Networks", *Computer Networks*, Vol. 2, 2003, pp. 269-284.

[4] Evits, P., "*A UML Pattern Language*", Macmillan Technology Series, 2000.

[5] Jayaratna, N., "*Understanding and Evaluating Methodologies, NIMSAD: A Systemic Framework*", McGraw-Hill, 1994.

[6] Jormakka, J, Jormakka, H., "State of the Art of Service Creation Technologies in IP and Mobile Environments", *Proceedings of IFIP World Computer Congress*, 2002, pp. 147-166.

[7] Larman, C., "*Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*", Prentice Hall, 2002.

[8] TINA-C, "*Service Architecture*", Version 5.0, 1997.