Using SDL for Modeling Behavior Composition

Jacqueline Floch\textsuperscript{1,2} and Rolv Bræk\textsuperscript{2}

\textsuperscript{1} SINTEF Telecom and Informatics  
N-7465 Trondheim, Norway  
jacqueline.floch@sintef.no

\textsuperscript{2} Department of Telematics  
Norwegian University of Science and Technology (NTNU)  
N-7491 Trondheim, Norway  
{\{jacqueline.floch,rolv.braek\}@item.ntnu.no}

Abstract. Behavior composition is a means to achieve modularity and adaptability. Unlike process algebra, SDL does not explicitly define composition operators. In this paper, we propose design patterns and rules for expressing elementary behaviors called roles and their composition in SDL. The composite state concept newly introduced in SDL-2000 is used in an original and innovative way to model roles and their composition. Simple SDL extensions are also discussed that facilitate composition. These extensions do not require changes to be made to the SDL semantics.

1 Introduction

Because a short time-to-market is increasingly important for new services, telecommunications operators and manufacturers are in constant search for better frameworks and methods for the rapid construction and deployment of services. Reuse, adaptation, dynamic composition and configuration are some techniques that can contribute to rapid service construction and deployment. At NTNU, the PaP project has explored these techniques in order to define a framework for service development and execution that enables services to be designed separately, and then composed dynamically using Plug-and-Play techniques \cite{1}.

In that project, our work has addressed the following question:

How can we model services so that they can be easily composed and adapted – possibly at run-time?

Composition and adaptation is simplified when services are designed in a modular way, and functionality can be reused. Services usually involve the interaction of several components, allowing adaptation to be performed at different levels. In a coarse-grained approach to adaptation, service components are replaced, or new components added. In a fine-grained approach, existing service components are partially modified. We have chosen to address a fine-grained approach, and seek to produce services by composing small behavioral elements in various ways. Our choice is inspired from existing service architectures such...
as IN (Intelligent Network), where a fine-grained approach has been successfully adopted [2], and TINA (Telecommunications Information Networking Architecture), that defines a set of service scenarios and interfaces as basic elements of a service [3]. Another reason for adopting a fine-grained approach is that small adaptations are typically required in the provision of customizable services to the mobile users.

We adopt a role based design approach [4]. Roles and role collaborations focus on behaviors across a system boundary. Experience suggests that role modeling provides better support for system adaptation and reuse than class modeling. The unit of reuse is seldom a class, but rather a slice of behavior [5]. In our approach, services are modeled as collaborations between functional roles. Complex roles may be decomposed into small elementary roles in order to break down behavior complexity. Conversely, more complex roles, and thus behaviors, can be produced by composition.

This paper presents our approach to service role modeling and composition using SDL [6]. Describing system behaviors in terms of state machines has proved to be of great value, and is widely adopted in most teleservice engineering approaches. We favor the use of the modeling language SDL because of its formal semantics that enables an unambiguous interpretation of the service specification. Using SDL, we are able to reason completely about service role behaviors and interactions between roles at the design level. There exist various types of dependencies between roles that constrain how they may be composed. Although SDL does not define explicitly any composition operators, we here use SDL to describe different composition classes. A set of SDL concepts is selected for the realization of role composition, and general guidelines are drawn out for the specification of the roles to be composed.

In this paper, we first introduce to the modeling concepts of actor and service role. The assignment of roles to actors is presented in Sect. 3. Sections 4, 5 and 6 present the composition approach. Design rules and patterns are proposed, and some extensions to SDL are introduced. The advantages of the composition approach are discussed in Sect. 7.

2 Actors, Roles and Collaborations

Service design is complex. Communication services normally require the coordinated effort of several distributed components that execute concurrently, and some components may be involved in several services. In a PaP context, this complexity even increases as services should be designed such that they can be dynamically adapted.

Our approach to service design makes use of roles [7, 8]. A service is seen as a collaboration between service roles where a role is the part a computational object plays in a service. A computational object that is involved in several services plays several roles that are composed to form the object behavior. By using roles, we are able to better comprehend the collaborations between the computational objects involved in a service, and to break down the complexity