Applicability of UML in Protocol and Test Development

Tamás Kasza, Sarolta Dibuz, Tibor Szabó, and Gyula Csopaki

In the last few years, UML (Unified Modeling Language) has become the dominant modeling language of the software industry. UML is also in the focus of the telecommunications industry, because – due to its flexible, platform-independent and generalized description features – it is ideal for modeling telecommunications systems. UML regards systems as models from several points of view. The most commons are use case, logical, component and deployment views. The application of the existing standardized description techniques and languages – such as SDL (Specification and Description Language) – is rather time consuming from the business and the technical perspective as well, because those methods are generally very complex for practical use. UML fulfils the needs and requirements of protocol development.

Particularly, UML is a visual modeling language for specifying, constructing, visualizing, and documenting the artifacts of a concurrent, distributed software-intensive system, and it does draw the line as developers move toward code.

In this paper, we investigate three potential ways for utilizing the UML in the field of communication protocol engineering. We discuss the advantages and drawbacks of the standardized mapping from UML to SDL, which is a language primarily used for specifying network protocols and possibly for describing digital logic. On the other hand, we raise the problem of reverse engineering. It is interesting from the aspect of implementation, that is, we investigate the SDL to UML mapping. Finally, we show how to derive conformance test suites from UML models of network protocols.

The standard ITU-T Z.109 introduces the relationship between some elements of SDL specifications and some UML diagrams. It presents the process of the requirement collection, analysis and draft design, which is necessary to formalize the plain text standard. Firstly, this paper discusses the differences and similarities between UML and SDL, investigates UML mapping based on Z.109, and proposes the use UML and SDL together in order to get advantages of both languages.

However, it is interesting to investigate the reverse direction, the use of UML, to create more fine models during the implementation phase. If UML is used in the implementation process as well, the product will have numerous favorable properties. UML provides adaptation to different object-oriented development environments used nowadays. Furthermore, UML meets the requirements of modularity and distributed systems according to the formal specification. This expressing semi-formal technique of modeling makes the process of mapping formal specifications to modern, object-oriented languages considerably easier than the direct mapping. This train of thoughts arises the question whether there is relationship between the UML models used before and after the formal description. Secondly, the article gives an overview on this reverse engineering process as well.

Thirdly, UML can also be used to support the test development for protocol specifications such as conformance test suites. After the identification of independent system components, like the tester(s) and the implementations under tests, test configurations can be developed using component and deployment diagrams. Test case structures and test purposes are defined for several test configurations. We consider the widely-used test description facilities Message Sequence Charts (MSC) and Tree and Tabular Combined Notation (TTCN) to be the target test notation.

Keywords: UML, SDL, MSC, telecommunications protocol specification and testing, TTCN