This section aims to provide a brief description of the SDL (Specification and Description Language) and illustrate the technique by applying on the go-back-n ARQ protocol. An overview of tool support and critical evaluation are then presented to summarize the section itself.

SDL is evolved and standardized following the development of object-oriented software programming approach. It is designed to be an effective and intuitive high-level general-purpose description language which specializes on real-time communication systems.

Based on the Extended Communicating Finite State Machine (ECFSM) technique which includes communicable concurrent processes with extended variables and data space, SDL has distinguished the structural and functional aspects into building blocks with interconnecting channels on the system level, and internal processes in accordance with their required operations on the behavior level. The extended variables and data space can communicate by exchanging control signals or messages on asynchronous channels. The association among SDL structural blocks and processes is accomplished by declaration of processes and interconnecting routes between processes and block’s interfaces in the block diagram. SDL processes are characterized by a finite number of control states connected by a finite number of transitions which can be triggered by external stimulus, timer timeout or validity of state variables. There is a set of tasks performed within each execution of transition. SDL also supports non-determinism to describe environmental uncertainty and validate possible system reactions.

Abstract Data Types (ADT) is used to characterize variable and parameter types through inheritance to avoid redundant redefinition of common operators and to define parametric data types.

Communication on bidirectional routes between processes is asynchronous in SDL, and synchronous communication is provided by the remote-procedure call instruction. Also a non-ordered signal reception is supported by SDL.

An example of SDL is presented to describe the Go-Back-N ARQ protocol which operates on a noisy channel. According to SDL, the system diagram decomposes the GBN system into three basic blocks: transmitter, channel and receiver. Each block is associated with related process represented by an octagonal shape. The process diagrams declare local variables to perform internal data handling. The SDL diagrams of the three protocol entities have specified and summarized their behaviors. The transmitter transmits PDU (Protocol Data Unit) as long as the ACK-based transmission window is below a certain maximum value with a default timer. The receiver checks whether the sequence number of the received PDU matches the expected sequence number, and issues ACK signal if positive judgment is made. In the system diagram, PDU and ACK signals are conveyed to the channel process with non-deterministic behavior of data handling.

Being the most exploited formal specification language in the field of telecommunication manufacture, SDL has supported many upper level software packages. There are promising freeware tools such as JADE and SITE, and also there are successful commercial products such as Telelogic Tau SDL Suite (SDT) and Object-Geode. Example of application can be found in the specification and simulation of UMTS Radio Access Network (UTRAN) based on the SDT software suite.

The modular approach and the clear distinction between structure and behavior of SDL are very useful features in describing OSI-like protocol architectures. However the lack of adequate freeware tool support for validation leads to alternative solutions to translate SDL into other formal notations.