

Specification and Description Language

9 languages

[Article](#) [Talk](#)

[Read](#) [Edit](#) [View history](#)

From Wikipedia, the free encyclopedia

Specification and Description Language (SDL) is a [specification language](#) targeted at the unambiguous specification and description of the behaviour of reactive and [distributed systems](#).

Overview [\[edit \]](#)

The [ITU-T](#) has defined SDL in Recommendations Z.100 to Z.106. SDL originally focused on telecommunication systems; As of 2016 its current areas of application include process control and real-time applications in general. Due to its nature it can be used to represent simulation systems ^[1] without ambiguity and with a graphical notation.

The Specification and Description Language provides both a graphical *Graphic Representation* (SDL/GR) as well as a textual *Phrase Representation* (SDL/PR), which are both equivalent representations of the same underlying semantics. Models are usually shown in the graphical SDL/GR form, and SDL/PR is mainly used for exchanging models between tools. A system is specified as a set of interconnected abstract machines which are extensions of [finite-state machines](#) (FSM).^[2]

The language is [formally complete](#),^[3] so it can be used for code generation for either simulation or final targets.

The Specification and Description Language covers five main aspects: structure, communication, behavior, data, and inheritance. The behavior of components is explained by partitioning the system into a series of hierarchies. Communication between the components takes place through gates connected by channels. The channels are of delayed channel type, so communication is usually asynchronous, but when the delay is set to zero (that is, no delay) the communication becomes synchronous.

The first version of the language was released in 1976 using graphical syntax (SDL-76). This was revised in 1980 with some rudimentary semantics (SDL-80). The semantics were refined in 1984 (SDL-84), the textual form was introduced for machine processing and data was introduced. In 1988, SDL-88 was released with a formal basis for the language: an abstract grammar as well as a concrete grammar and a full formal definition. The version released in 1992 (SDL-92) introduced object-oriented concepts such as inheritance, abstract generic types etc., with the object-oriented features described by transformations into non-object oriented ones. SDL-2010 is the latest version, an updated version of SDL-2000 that was completely based on object-orientation, rather than description by transformations. This version is accompanied by a [UML-Profile](#): ITU-T Recommendation Z.109 (04/12), SDL-2010 combined with UML. SDL-2010 also introduced the support of [C data types](#) as initially introduced by SDL-RT.

Organization [\[edit \]](#)

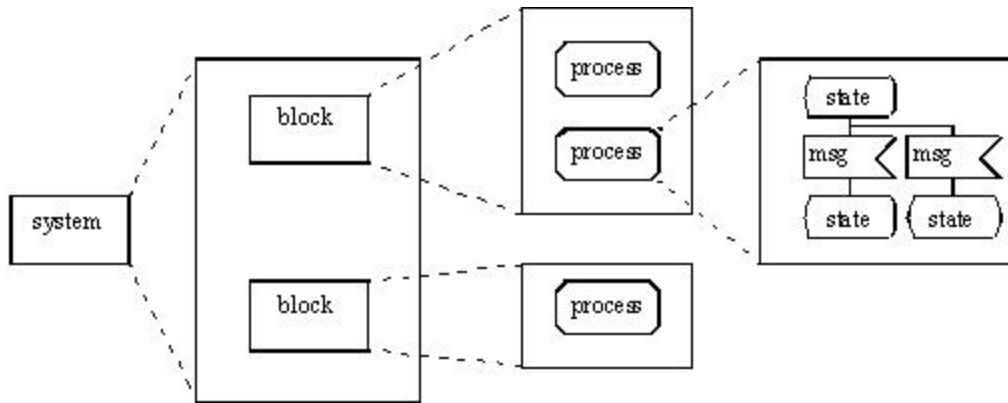
Hierarchy level [\[edit \]](#)

The hierarchy level of SDL is structured as follows.

- Library package
- System agent
- Block agent
- Process agent
- Procedure type
- Remote procedure

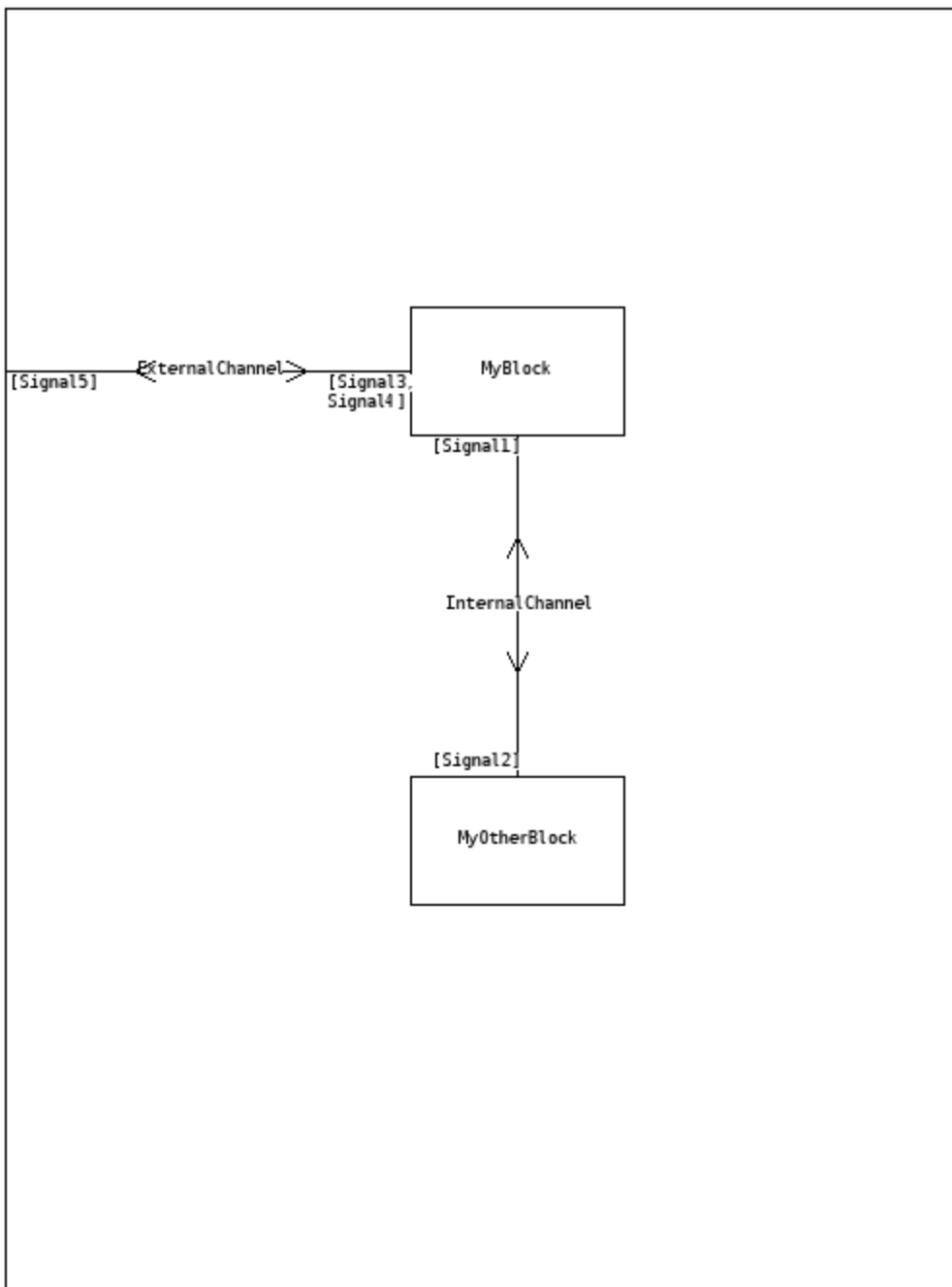
Architecture [\[edit \]](#)

An SDL system is made of functional blocks and each block can be further decomposed in sub-blocks. The lowest level block is composed of one process or several processes described as finite-state machines.



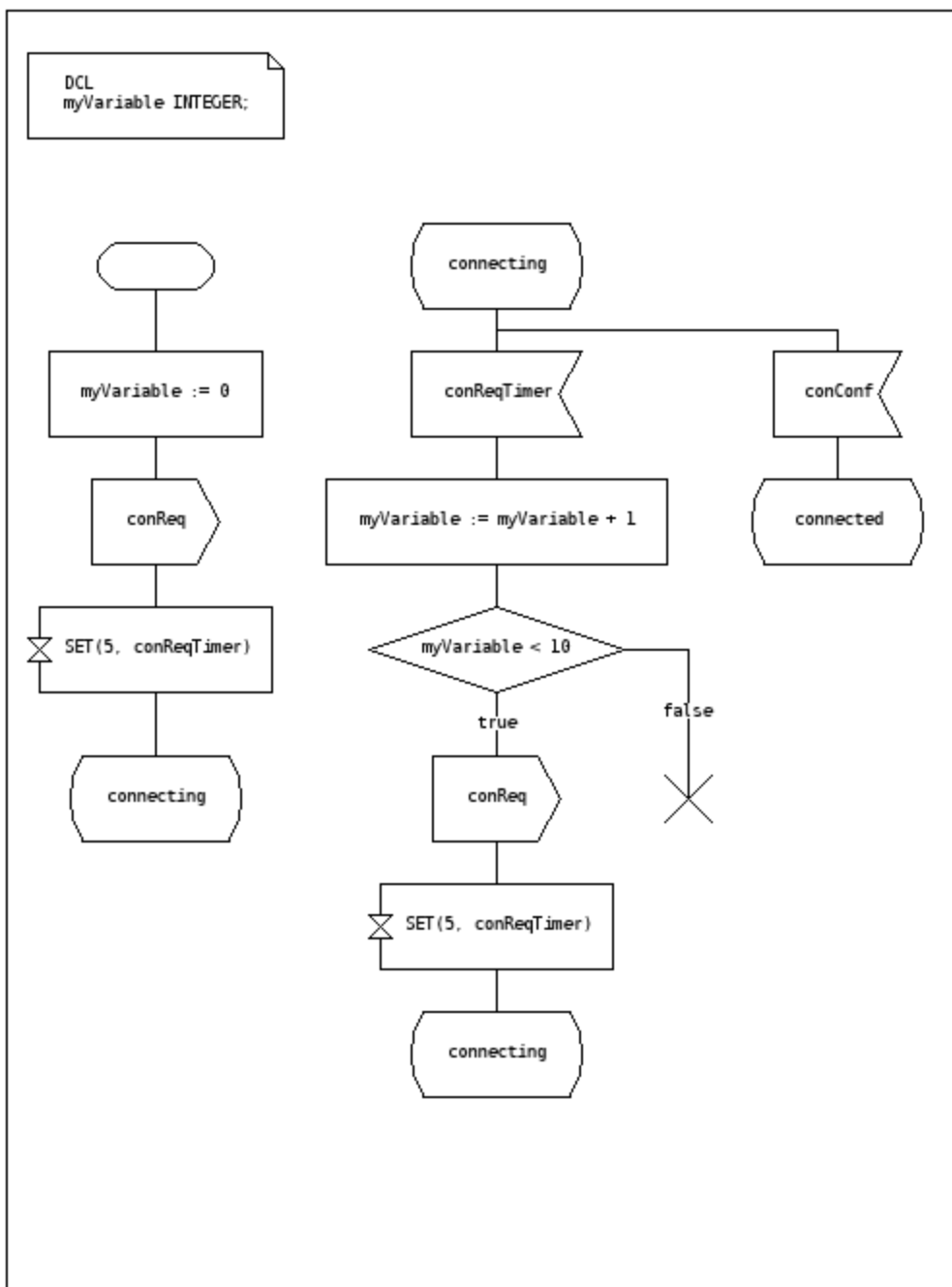
Communication [\[edit \]](#)

Blocks are connected through channels that carry the messages (or signals) exchanged between the blocks. A block agent consists of process agents.



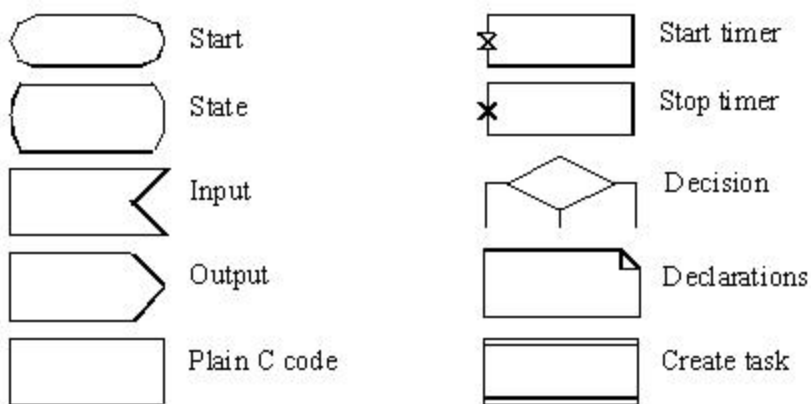
Behavior [\[edit \]](#)

Each process agent is a state machine that contributes to the action carried out by the system. A message stimulus coming from the environment or from another agent to an agent is called a signal. Signals received by a process agent are first placed in a queue (the input port). When the state machine is waiting in a state, if the first signal in the input port is enabled for that state it starts a transition leading to another state. Transitions can output signals to other agents or to the environment. A process agent is allowed to contain procedure types so that the same actions can be invoked from different places. It is also allowed to call a remote procedure type to invoke a procedure in another agent (or even another system) and wait for a response.



In this example MyVariable is of type **INTEGER** and is the only variable in the process. The first transition is the *start* transition that initializes the **local variable**. A connection request message *conReq* is sent, a 5 seconds timer *conReqTimer* is started, and the state machine goes to the *connecting* state. In the *connecting* state if the timer goes off -that is equivalent to a message receive- the connection request is sent again up to 10 times. If a connection confirmation is received the state machine goes to *connected* state. This is a typical telecommunication protocol scenario.

Available symbols are:



Data [\[edit \]](#)

SDL Abstract Data Types (ADT) support basic data types such as INTEGER, REAL, CHARSTRING as well as structured ones such as structures (STRUCT), enumerated (LITERALS), constants (SYNONYMS).

Syntax looks like the one from Pascal, for example an assignment is written ':='.

Variants [\[edit \]](#)

- SDL-RT is based on the ITU Specification and Description Language replacing the data language with C.^[4] Latest version of SDL standard (SDL'2010) now includes the support of C data types and syntax making SDL-RT basic principle part of the official standard.
- [TNSDL](#) is a variant developed by Nokia for their internal use.

Related technologies [\[edit \]](#)

- MSC [Message Sequence Chart](#). Another ITU-T related notation to describe the dynamic behavior of a system.
- [ASN.1](#) (Abstract Syntax Notation One). An ISO standard to describe data types that can be used within SDL.
- [TTCN-3](#) Testing and Test Control Notation version 3. A testing language standardized by ITU-T and written by ETSI.
- [Unified Modeling Language](#) A generic modeling notation. A profile has been standardized based on SDL.

See also [\[edit \]](#)

- [Specification \(technical standard\)](#)
- [Specification language](#)
- [Ivar Jacobson](#)

References [\[edit \]](#)

- ↑ Fonseca i Casas, Pau and Pi, Xavier and Casanovas, Josep and Jové, Jordi (2013). "Definition of Virtual Reality Simulation Models Using Specification and Description Language Diagrams". *SDL 2013: Model-Driven Dependability Engineering*. Lecture Notes in Computer Science. Vol. 7916. [Springer Berlin Heidelberg](#). pp. 258–274. doi:10.1007/978-3-642-38911-5_15 . hdl:2117/22156 . ISBN 978-3-642-38910-8.

2. ^ This article is based on material taken from [SDL](#) at the [Free On-line Dictionary of Computing](#) prior to 1 November 2008 and incorporated under the "relicensing" terms of the [GFDL](#), version 1.3 or later.
3. ^ R. Grammes; R. Gotzhein (2007). *Fundamental Approaches to Software Engineering*. Lecture Notes in Computer Science. Vol. 4422. [Springer](#). pp. 200–214. doi:10.1007/978-3-540-71289-3_17 . ISBN 978-3-540-71288-6. S2CID 22556123 .
4. ^ "[Specification and Description Language – Real Time](#)" . Retrieved 2014-03-29. "SDL-RT prime use is to develop real time and embedded software. SDL-RT is based on tasks running concurrently that exchange information through messages and synchronize with each other using semaphores."

External links [[edit](#)]

- [ITU Languages and general software aspects for telecommunication systems](#)
- [SDL Forum Society](#)
- [List of SDL tools](#)



Wikimedia Commons has media related to ***Specification and Description Language***.

Authority control databases

[hide]

National United States · Israel

Other Yale LUX

Categories: Formal specification languages | ITU-T recommendations | Diagrams
Software modeling language | Telecommunication protocols

This page was last edited on 21 April 2026, at 17:05 (UTC). Page was rendered with [Parsoid](#).

Text is available under the [Creative Commons Attribution-ShareAlike 4.0 License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Legal & safety contacts](#) [Code of Conduct](#) [Developers](#) [Statistics](#)

[Cookie statement](#) [Mobile view](#)

--	--