

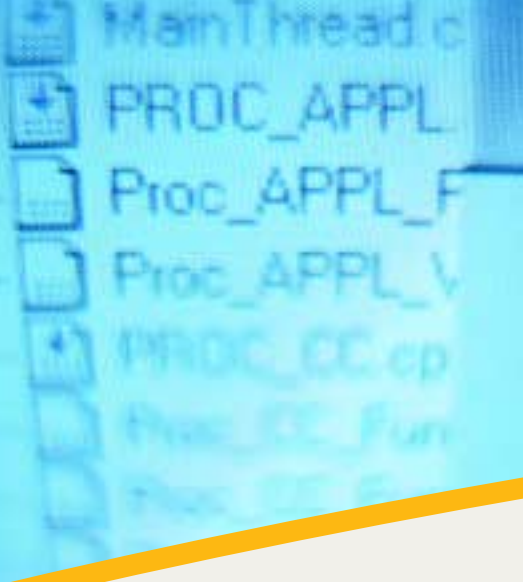


mobile communication engineering

Graphical Software Design



SDI Sphere



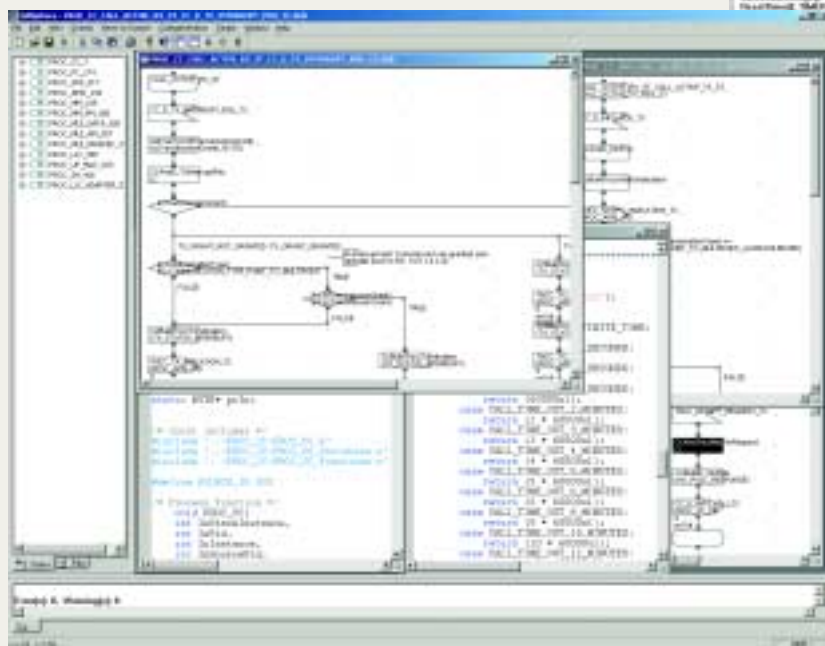
SDL or the Specification and Description Language is originated in the field of telecommunications. The language was developed and standardized in this field in order to describe complex systems in a graphical way. SDL combines all phases from system design to detailed model specification to actual implementation. This resulted in an established and widely used common standard.

**SDL Sphere** can be used for the **design of any kind of FSM or Finite State Machines**. Since **compatible source code** (e.g. ANSI-C) is generated the **easy integration into any target hardware** is possible.

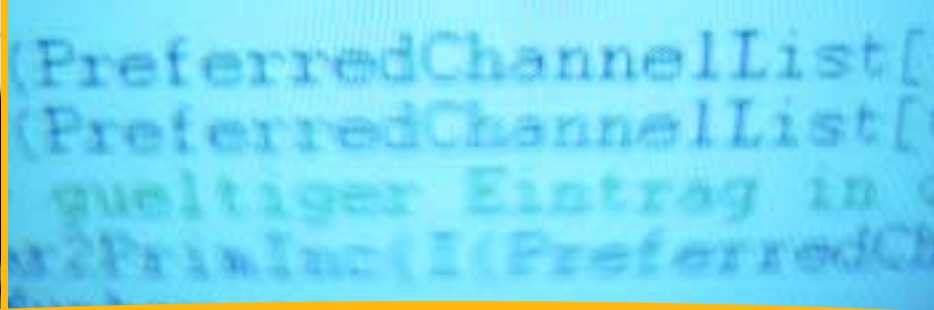
**SDL Sphere** combines not only the graphical editor for the SDL sheets including navigator but also the **code generation** in a joint IDE or Integrated Development Environment. The integrated code generator generates **compact, transparent and fast source code**, which can be used in a very flexible manner.

### Areas of deployment:

- Telecommunication systems
- Protocol stacks
- Machine controls
- Driver stages
- Real time systems of any kind
- Generally any state-based software

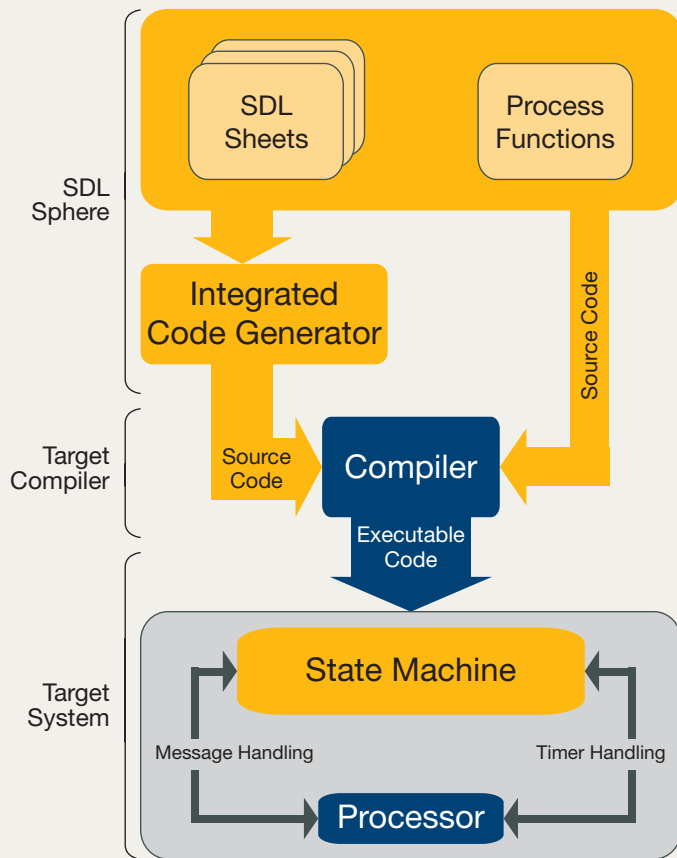


SDL Sphere



## Architecture:

The programming in SDL can commence right after creating the rough system design. The automatic code generation uses the SDL sheets created by you and generates the required source code. This code requires only one adaption to the target system. The whole additional development is carried out in SDL Sphere.



Editing the SDL sheets and additional files and functions in the target language is carried out using this interface.

The code generator in SDL Sphere works like a pre-compiler used before the actual target compiler. The development in two languages is used for this reason, since this combines the clearness of SDL with the required functionality of the target language.

The compiler of your target system transfers the state machine to the target. The interfaces for the message and timer handling have to be adapted only once and then it is possible to create additional complex systems quickly. These systems can be used at once.

By using the graphical concept the clearness of the software is maintained during the entire development cycle.

You automatically receive the documentation which is also the basis of the source code. The system design and documentation are united and cannot drift apart anymore.

Even very complex systems stay clear. This facilitates your work in regard to the entire project and in your team.

## Outline:

- Cost-effective, complete SDL IDE. Supplies all necessary tools.
- Optimized and therefore fast code, especially for real time applications; therefore reduction to the essential SDL contents.
- Universally applicable to any state machine on any target system.
- Transparent, readable code.
- Directly integrated documentation by means of the philosophy of the SDL Sphere.
- The flexibility of the target language is not restricted.

# Benefits





## Support:

During the first SDL projects an integrated tool will offer support to the user. This tool creates a complete and translatable Microsoft Visual C++™ project. In addition, we offer prepared state machines (e.g. an elevator control) which will support you in getting to know the environment quickly and goal-directed.

Our examples will show how traces of a state machine can look like and how a complete SDL system can be simulated using Windows. You can also directly create an executable project with your own state machine.

Aside from the code generator for ANSI-C we also offer generators for JAVA™, C#, etc.

## In addition:

- Extensive user guide and numerous examples facilitate the handling of the SDL Sphere.
- Additional examples illustrate possible integrations in target systems such as Unix®, ThreadX®, QNX®.
- We offer support for the integration into your target system.
- Training for SDL and system design are also available.

## Requirements:

- Common PC with a current version of a Microsoft Windows® operating system.
- Free USB port.
- Minimal operating system (message, timer handling) and compiler for the target hardware.

There are hardly any requirements regarding the target system. This is where SDL Sphere demonstrates its flexibility. The integration into the target hardware is quickly achieved and can also be carried out parallel to existing state machines.