Industrial robots are used in different application fields. Many manufacturers are active on the market with their robots and software for robot programming. In most cases the software solutions are manufacturer-specific and cannot be used for other robots. Long term standardization efforts for explicit robot programming, such as the IR (Industrial Robot Language) and its successor, the PUR (Programming Language for Robots), have been mostly ignored by robot manufacturers. This fact leads to a restriction on the continued usage of robots from different manufacturers or makes it even impossible. Multirobot applications, where robots have to interact, are mostly limited to the software solution and robots of one manufacturer.

Control model design in the engineering and scientific domain is often characterized by the usage of Scientific and Technical Computation Environments (SCEs) like MATLAB. The Robotic Control & Visualization (RCV) Toolbox for MATLAB closes the gap between robot manufacturer-specific programming languages and SCEs. The current version of the RCV Toolbox for MATLAB supports control development for KUKA and KAWASAKI robots. However, an extension to other robots is straightforward.

Research in robotics is proceeding rapidly and new fields of application for robots are made up continually. The requirements concerning robot control development are increasing, too. Easy programming and integration of various external hardware (sensors, actors) are of particular importance. In this context, it is desirable to have a homogeneous software environment from the early control design to operation phase (Rapid Control Prototyping). The RCV Toolbox for MATLAB provides an efficient way to realize these requirements in practice.

- One Control-Program for interactive robots of different manufacturers
- Easy, manufacturer independent integration of external hardware

The RCV Toolbox distinguishes between three object types. The central part are objects of type ROBOT which are interacting with objects of type PARTS. ENVIRONMENT objects are passive and cannot be moved by a ROBOT object.

The SBC framework is a computer aided development methodology for discrete event controls. The SBC is based on the concept of Rapid Control Prototyping (RCP). An essential aspect is the continuous usage of simulation models during the entire control program development.

Some basic control commands of RCV Toolbox:

- rkill(...) — Stops real robot
- rmove(...) — Starts movement of robot
- rpoint(...) — Places robot at position
- rprozess(...) — Processes real robot
- rreset(...) — Restores initial robot state

A model is gradually expanded in a control program in operation and continually tested by simulation. This kind of integrated software development requires a software chain (tool chain). Moreover, it is necessary to distinguish as early as possible between the Control Model (CM), the process model and the Interface Model (IM) in order to get an overview of the behavior of real processes and devices. The operation phase is a communication with the real process needs is needed. The Interface Model (IM) provides this communication and supports an interaction with real and virtual process devices. The RCV Toolbox is used to implement the entire CM and parts of CIM and PIM.

Some basic visualization commands of RCV Toolbox:

- VirtualRobot.delete_all
- VirtualRobot.place_env
- VirtualRobot.repose_robot
- VirtualRobot.place_part
- VirtualRobot.repose_env
- VirtualRobot.repose_part
- VirtualRobot.delete_all

The application consists of three interacting robots using geometric coupling (load sharing). After simulation testing of the control program, the virtual robots can be replaced by real ones without any change in the control program.