

CHAPTER 6.

THE MAIN PROGRAM

A discrete event system description is written formally as a SIMULA block.

SIMULA begin ----- end

A SIMULA block is itself a statement, which may be a part of an otherwise ordinary ALGOL program. The SIMULA concepts introduced in the preceding chapters are only available within a SIMULA block. A program may contain more than one SIMULA block; however, nested SIMULA blocks are not permitted.

The SQS of a given system is local to the SIMULA block. The SIMULA block head must contain all activity declarations in the system. Activity declarations are not permitted elsewhere. Element and set declarations can be given anywhere inside a SIMULA block, i.e. also within sub-blocks of any kind, including procedures and activities.

The SIMULA block functions dynamically as a process, which will be called the main program. It is always present in the system. The initial system setup is conveniently done in the main program. All sequencing statements have their usual significance.

Upon entry into a SIMULA block the main program is the currently active process, and the current system time is equal to zero. The SQS contains one event notice referencing an element with no SM, which in turn refers to the main program.

The apparent paradox that the main program should function both as a process in parallel with other processes and as an outer block to all processes (therefore to itself), is resolved by defining the outer block to a process to be a connection block connecting the main program (i.e. the SIMULA block). The fact that the latter has itself connected in an immediately outside block causes no difficulty. The connection is through the element mentioned above.

A go to statement leading from within the body of an activity, i.e. from a process, into the main program has the usual effect of a go to statement leading into a connected process (see section 5.2).

When leaving the SIMULA block through its final end the simulation is terminated, even if further events have been scheduled. The same is true if a go to statement is executed, leading out of the SIMULA block.

If the SQS becomes empty before exit from the SIMULA block is made, further actions are undefined. This is possible only if all processes, including the main program, are passive or terminated.

The statement `terminate(X)`, where X refers to the main program, will not terminate the simulation. Normal termination can still be achieved by executing a go to statement leading to a suitable point in the main program or leading out of the SIMULA block.