CHAPTER 2.

PROCESSES

2.1 Basic Properties.

The process concept is fundamental in SIMULA. It serves to organize and classify the actions taking place in a discrete event system as well as the data involved. A discrete event system is viewed as a collection of processes, whose actions and interactions completely describe the operation of the system.

Processes enter and leave the system as the result of actions performed within the system itself. During its presence in he system a process must be regarded as an individual entity distinct from other processes. It has its own local data and its own behaviour pattern.

A system may contain several processes with a similar data structure and the same behaviour pattern. Such processes are said to belong to the same class, called an activity.

A process is described by an activity declaration, which is common to all processes belonging to that class. An activity declaration has the same syntax as a procedure declaration in ALGOL, except that the first symbol of the heading is "activity". There is no "<type> activity" concept. (See also section 2.4)

A reference to an activity is called a process designator (see section 3.4.1). It invokes the generation of an individual process described by that activity declaration, in which the formal parameters are replaced by the specified actual ones. The process designator is itself an expression referencing the generated process.

The process will usually remain in the system after the evaluation of the process designator. Its actions are executed in parallel with those of other processes in the system, in the sense that sequences of actions of different processes are interlaced.

The data structure belonging to a process are the data local to it, as defined in the activity declaration. The statements of the activity body constitute the operation rule of the process. The body can have an arbitrary block structure.

The items local to the outermost block of the body are called endogenous attributes; the formal parameters are called the exogenous attributes. The attributes of a process can be made accessible to the outside, i.e. to other processes, see chapter 5.

As the result of evaluating a process designator control enters the specified activity declaration. Thereby a local data block is created, which is distinct for this particular process. Two processes belonging to the same activity may differ in the values of their local variables, and they may be in different stages of execution at a given time.

2.2 The sequence control.

The actions performed by a process can be widely separated in time. They are grouped together in active phases separated by periods of inactivity. Inactive periods are invoked by certain sequencing statements, which are described in chapter 4. During such a period other processes are allowed to operate. However, during an active phase the process has complete control; there is no "interrupt" mechanism in the language.

A process can be viewed as a selfcontained program having its own local sequence control. When a statement invoking an inactive period is executed, the local control will stay at the end of this statement during that period, and will proceed to the dynamically next statement at the time of the next active phase.

The main sequence control applying to a SIMULA program as a whole, will go from one process to another and execute one active phase at a time, in a sequence defined by certain scheduling statements. Such a statement will schedule an event normally to happen at some later time. An event is the execution of the next active phase of a specified process.

At the end of an active phase of a process the main control leaves that process to execute some other event. However, a reactivation point is defined for the process, telling where in the operation rule the next active phase shall start. Exceptions are the cases when control leaves the process through the final end of its operation rule or by a go to statement. In these cases no reactivation point is defined. However, the process may remain in the system as a data structure.

We notice that the local sequence control of a process is identical to the main control during active phases, and it is represented by the reactivation point during inactive periods.

2.3 States.

A process can be in one of four possible states, depending on whether a reactivation point is defined, and on whether an event has been scheduled for it and not completed. The states are defined in the following table.

States	react. pt.	event
active	no	yes
suspended	yes	yes
passive	yes	no
terminated	no	no

Immediately after the generation a process is passive with a reactivation point on the first statement of its operation rule. The evaluation of value parameters and subscript bounds of endogenous array attributes is part of the evaluation of the process designator.

2.4 Exogenous attributes.

The parameter (exogenous attribute) mechanism of an activity declaration is less general than that of a procedure declaration. All parameters, except arrays, are called by value. The call by value is implicitly understood, and value specification does not apply. Label, switch, and procedure parameters are not accepted.

The parameter mechanism is such that a process can not in general refer back to the block in which the corresponding process designator was evaluated. Therefore a process may remain and operate after the latter block is out of the system, i.e. the life spans of different processes may overlap each other in any way. A process may, however, refer directly to items non-local to itself, i.e. items local to the block containing the activity declaration (see CHAPTER 6), or to outer blocks.

An exogenous array attribute is assigned as a local name on the array designated by the actual parameter of the process designator. It follows that an array may have a life span exceeding that of the block in which it was declared.