

The Birth of Object Orientation: the Simula Languages

by

Ole-Johan Dahl

Dept of Informatics, Oslo University

Talk given at Software Pioneers' Conference, Bonn 28-29/6 2001.

The Simula project

The project was initiated in 1962 by Kristen Nygaard, who saw the need for a language for simulation modelling.

From 1963 he and I cooperated very closely on language development. Implementation issues were my responsibility and issues of economy and organisation were KN's. The work took place at the Norwegian Computing Center (NCC).

General purpose algorithmic capability would be needed. Strategic considerations told that our language must be based on a standard one. Algol 60 was chosen.

There were three stages of language development, here identified as:

- Simula 0 (1962-63),
- Simula 1 (1963-65), and
- Simula 67 (1966-67).

2

Simula 0, 1962

The conceptual framework was mainly due to KN: a model would consist of "customers" flowing through a fixed network of service "stations".

Since both customers and stations had a superficial similarity to Algol blocks, the initial plan was to exploit that fact through a preprocessor to an Algol compiler.

Unfortunately the strict LIFO structure of Algol program executions is unsuitable for simulation models and could not be circumvented. For that reason the approach was abandoned.

4

Simula 1, 1963-64

The concepts of "customers" and "stations" were unified as "processes", able to operate in quasi-parallel over system time.

Processes were referenced by pointers (indirect through elements of circular lists).

They would admit access from outside to their "attributes", i.e. quantities declared in the outermost block.

7

Simula 1, 1963-64

For programming security the use of pointers had to be catered for by reference counts, later supplemented by a garbage collector.

Secure attribute access through unqualified pointers necessitated compiler conscious run time testing:

```
inspect <process reference> when <process type> do  
  begin <attributes accessible> end
```

8

Simula 1, 1964

The Algol language was extended by the following ad hoc mechanisms:

- Procedure-like **activity** declarations giving rise to quasi-parallel processes (only parameters called by value were allowed),
- *time*, a function procedure returning the system time,
- mechanisms for the explicit scheduling of processes in system time,
- attribute accessing by **inspect**, and
- a built-in concept of circular lists containing process pointers.

9

Simula 1, 1965-66

Our experience with customer applications showed that Simula 1 was indeed a useful tool for simulation modelling. But there was frustration also:

- A simpler **class** concept of objects not dependent on simulation oriented mechanisms (but able to operate like co-routines) could be important in a general purpose setting.
- The same would hold for list processing facilities based on simple object pointers.
- The **inspect** mechanism was clumsy at times.
- C.A.R. Hoare had proposed (in an Algol Bulletin) a scheme for "record handling" allowing direct access to record attributes based on pointers qualified by record class, and possibly by record subclasses.
- The Simula 1 compiler was based on an Algol compiler for the UNIVAC 1107 far from optimal for our kind of extensions.

13

Simula 1, 1965

The following skeleton example is found in the language documentation:

```
SIMULA begin;  
activity car; .. traveling on a linear road  
begin real X0, T0, V; .. position at given time, velocity  
  real procedure X; .. the position now  
    X := X0 + V * (time - T0);  
  procedure UpdateV(Vnew); real Vnew; .. change velocity  
  begin X0 := X; T0 := time; V := Vnew end;  
<car behaviour> end of car;  
activity police;  
begin ...; inspect <process reference> when car do  
  if X <within city> and V > 50 then UpdateV(50); ...  
end of police;  
... .. end of simulation model;
```

11

Simula 67, 1966

A new project for the development of an improved version of Simula was established, funded by the NCC.

A condition was that the project must prove profitable within the typical life time of a programming language, estimated to 3 to 5 years.

A new Algol/Simula compiler was planned in cooperation with The Technical University of Trondheim.

14

Simula 67, 1967

A main problem for us was how to adapt the class/subclass concepts of Hoare for our purposes. The breakthrough came in January '67, just in time for writing our paper for the forthcoming IFIP Conference on Simulation Languages at Lysebu, Oslo, May '67 ("Class and Subclass Declarations", in Ed.: J. Buxton: Simulation Languages, North Holland Publ, 1968).

A mechanism of class "prefixing" was defined, today referred to as "inheritance". The prefix of a class was itself a class, which could be separately compiled and reused.

15

For more, read: Stein Krogdahl: The Birth of SIMULA



<https://tinyurl.com/simulabirthkrogdahl>