An Object Oriented Journey

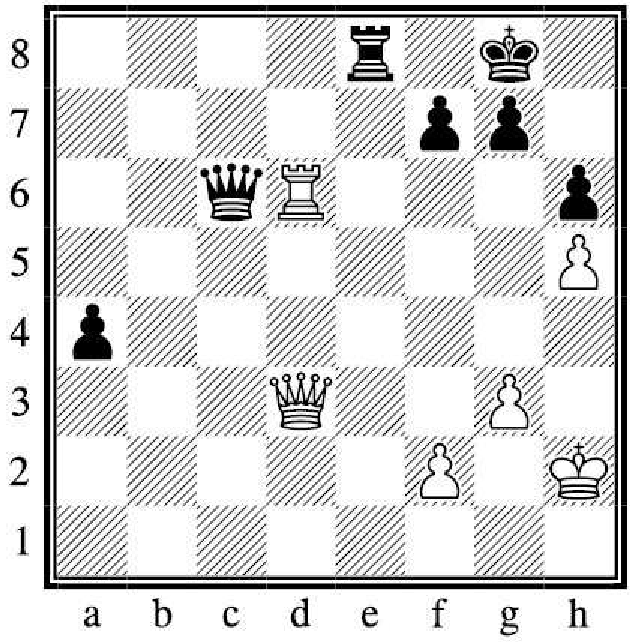
Georg Philippot

How Simula Shaped Me as a Programmer,
and Why I Moved on to Other Languages

1969

ALGOL on UNIVAC 1100

- My first complex application: Chess program

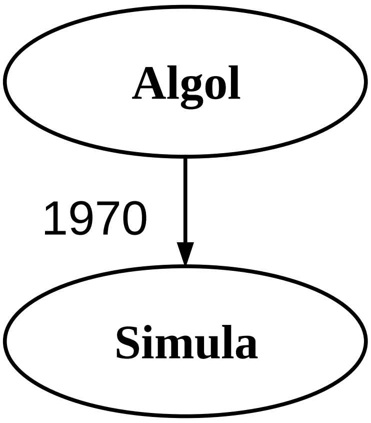


− 24 hour wait for next move

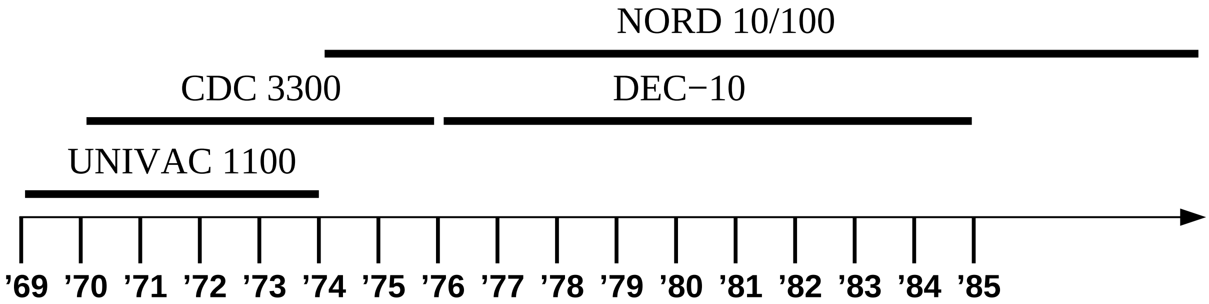
+ Some moves were smarter than expected and designed

− but they were caused by bugs in the program ...

And so the road to object oriented programming begins:

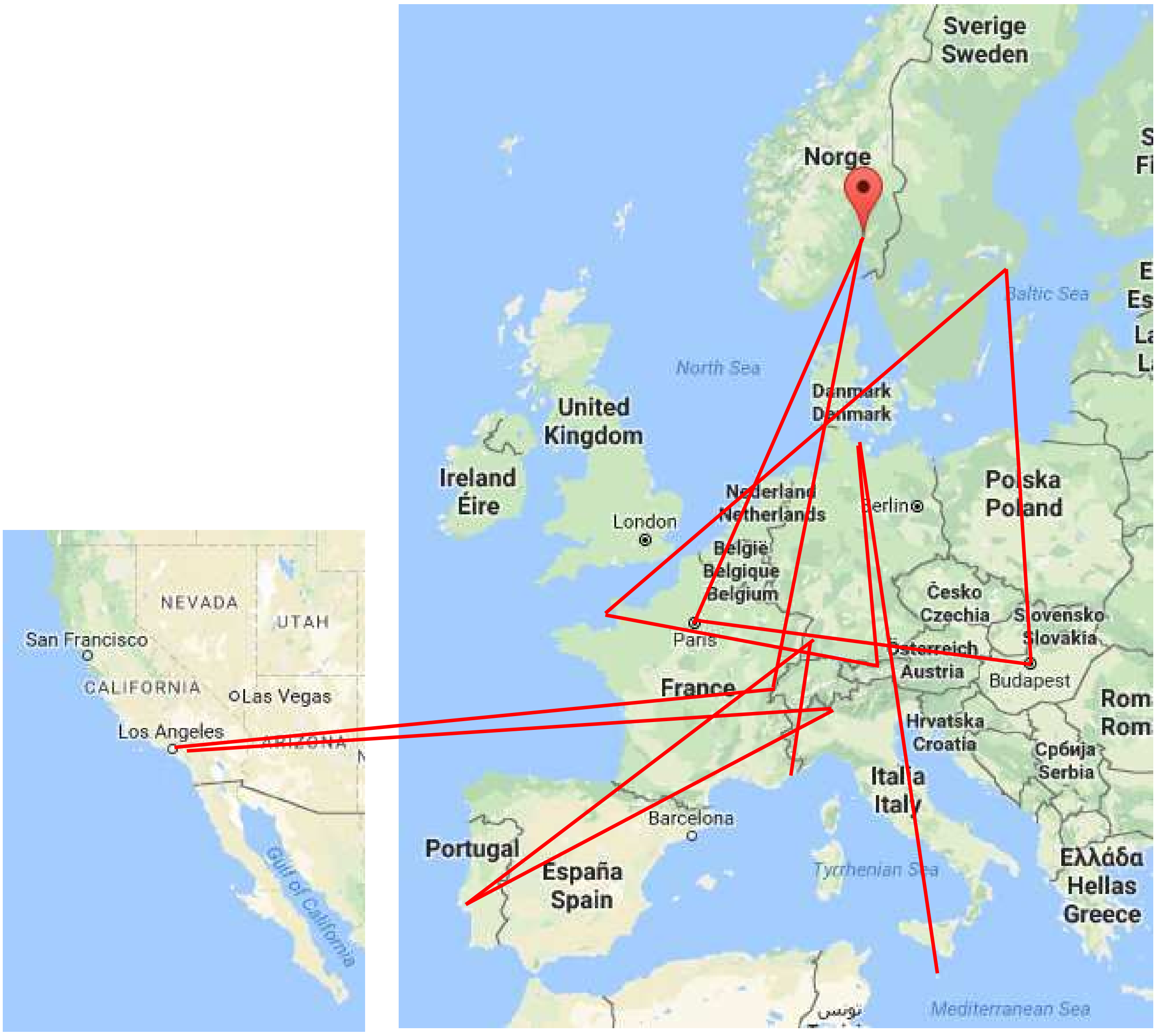


- Learnt and practised Simula on CDC 3300, UNIVAC 1100, later DEC 10 and NORD 10
- Very fascinating language; I used to daydream about objects while waiting for results of my program runs
- Taught the language to fellow students at NTH



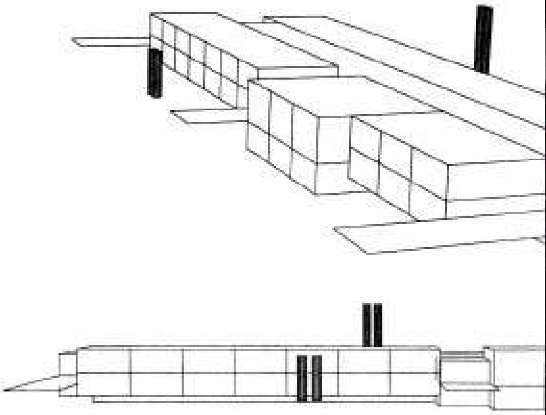
Early Application Phase

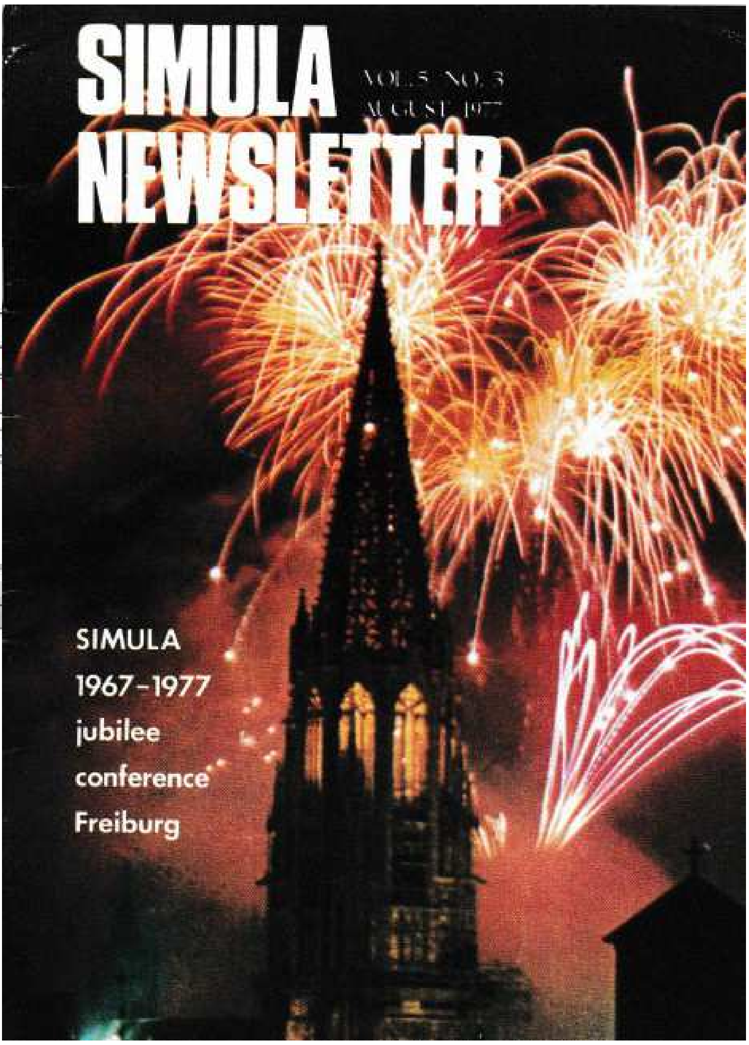
1973: ASU − Association of Simula Users



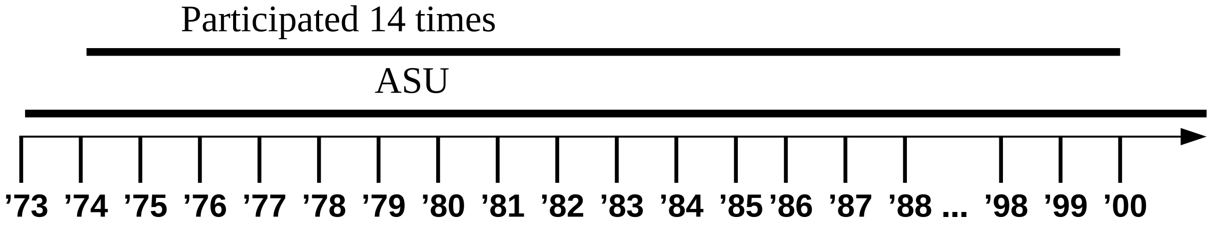
Early Application Phase

- PEPS − Perspective Processing System presented 1974 at the Monte Carlo ASU conference



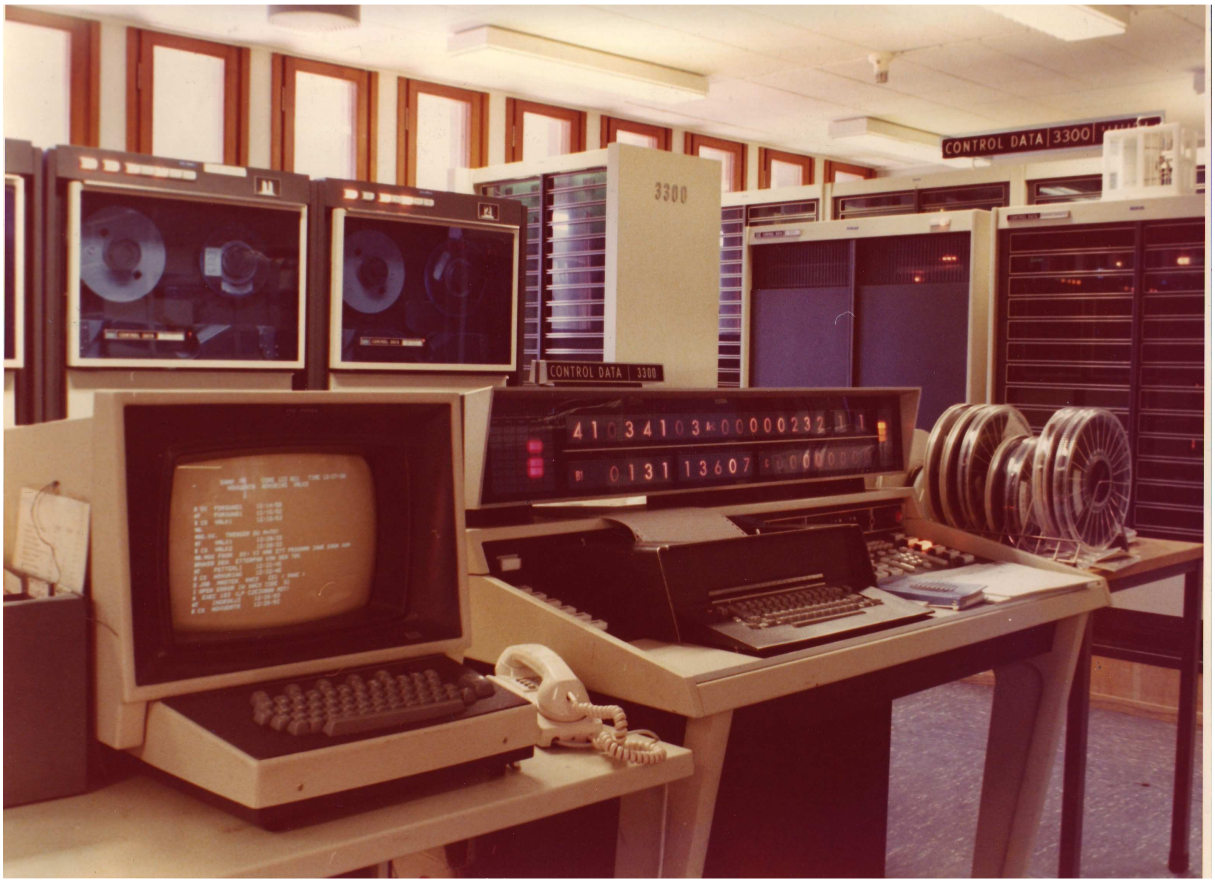


- Simula for NORD 10 presented 1977 at the Freiburg ASU conference
- Real time Simula presented 1980 at the Budapest ASU workshop on Simula and industrial systems


Participated 14 times

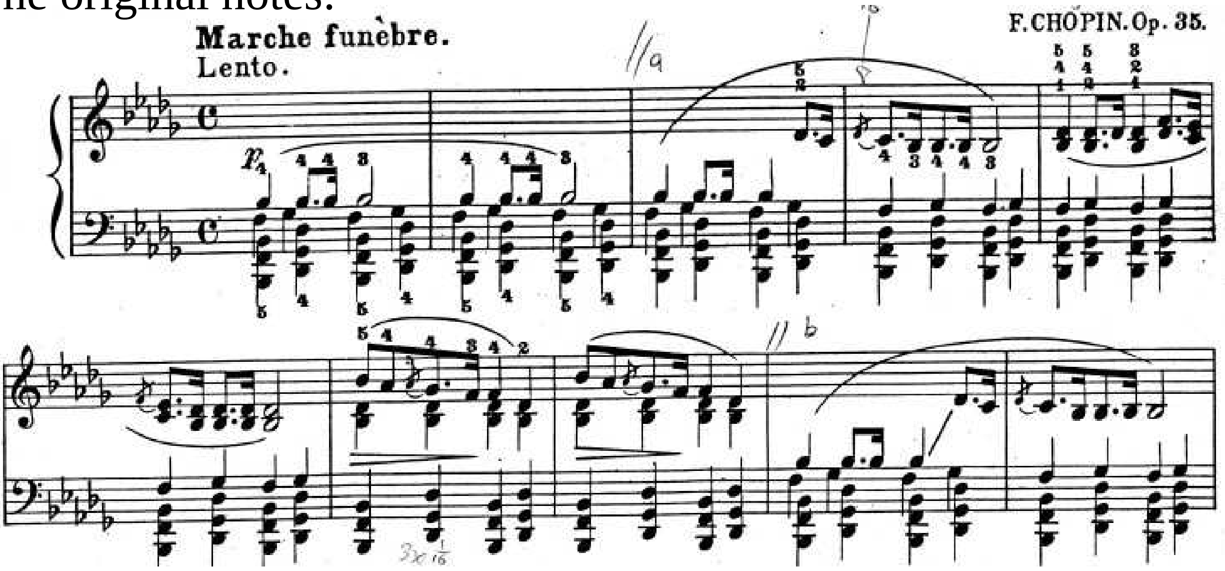
ASU

1976: Decommissioning the CDC 3300 at the University of Oslo

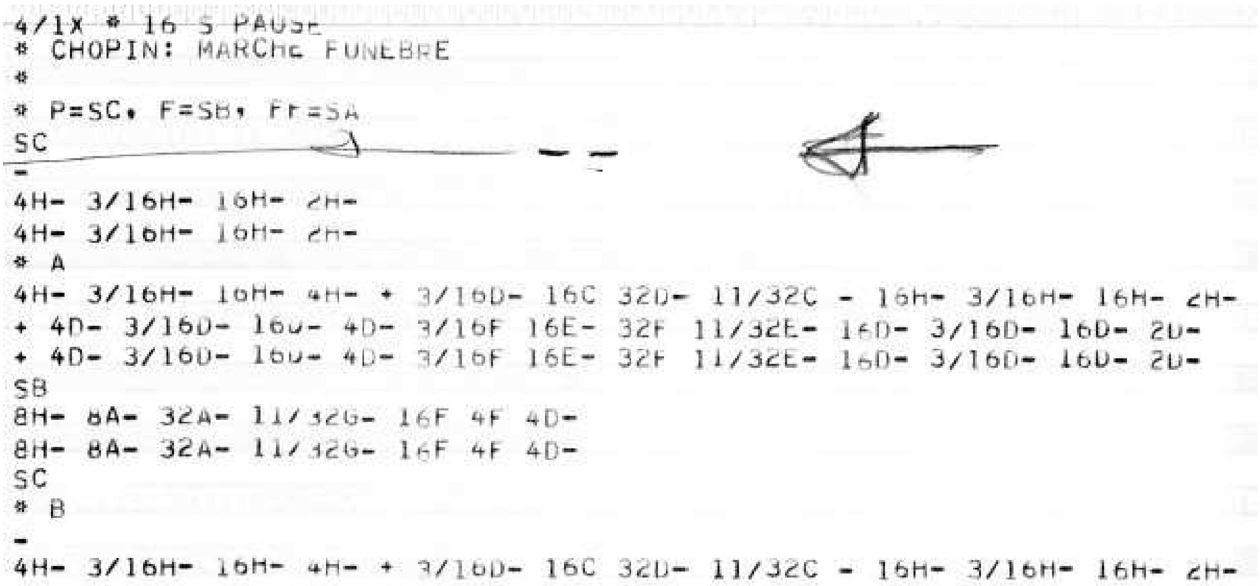


- I wrote the Simula program "music" which reads the notes from punched cards and turns the console into a synthesizer
- This was the last program to run on the CDC 3300, on June 11th, 1976
- The last tune played was Chopin’s funeral march − with all lights dimmed and the audience respectfully quiet

The original notes:



The punched cards:



```
4/1X * 16 S PAUSE
* CHOPIN: MARCHE FUNEBRE
*
* P=SC, F=SB, FF=SA
SC
-
4H- 3/16H- 16H- 2H-
4H- 3/16H- 16H- 2H-
* A
4H- 3/16H- 16H- 4H- + 3/16D- 16C 32D- 11/32C - 16H- 3/16H- 16H- 2H-
+ 4D- 3/16D- 16D- 4D- 3/16F 16E- 32F 11/32E- 16D- 3/16D- 16D- 2D-
+ 4D- 3/16D- 16D- 4D- 3/16F 16E- 32F 11/32E- 16D- 3/16D- 16D- 2D-
SB
8H- 8A- 32A- 11/32G- 16F 4F 4D-
8H- 8A- 32A- 11/32G- 16F 4F 4D-
SC
* B
-
4H- 3/16H- 16H- 4H- + 3/16D- 16C 32D- 11/32C - 16H- 3/16H- 16H- 2H-
```

Early Application Phase

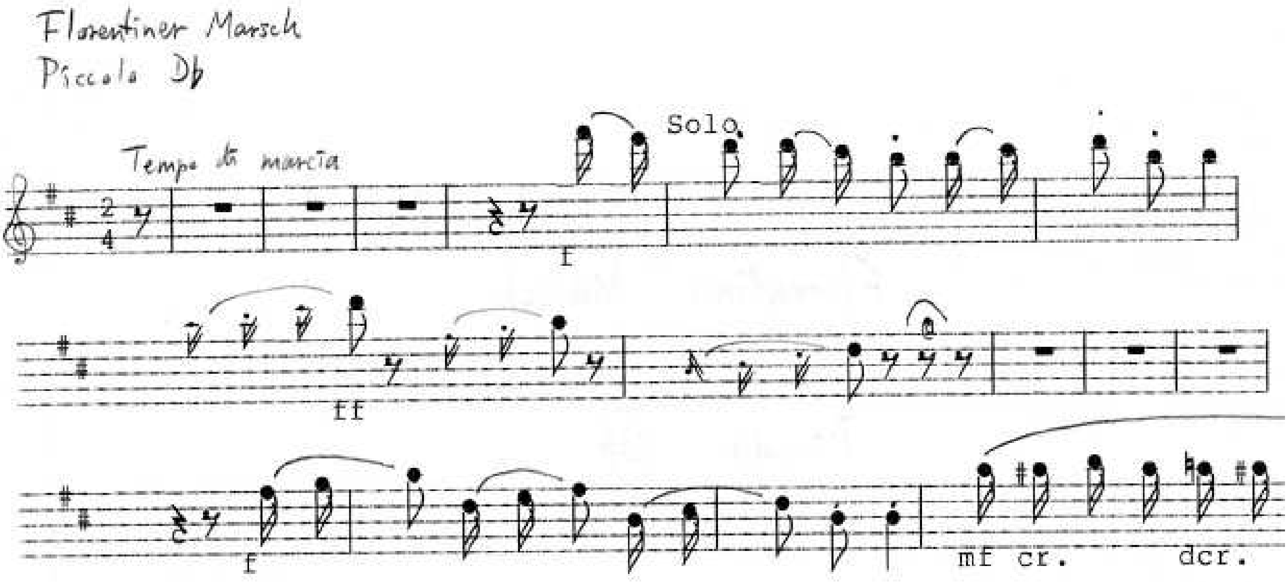
1977:

- SQAT − an interactive text editor
- War chess game
- Flight simulator game
- Games − why write them?
  - More fun to write games than to play them
  - Very useful when testing for bugs in the underlying Simula implementation

  The games were for two players and used semigraphics on two Tandberg screens

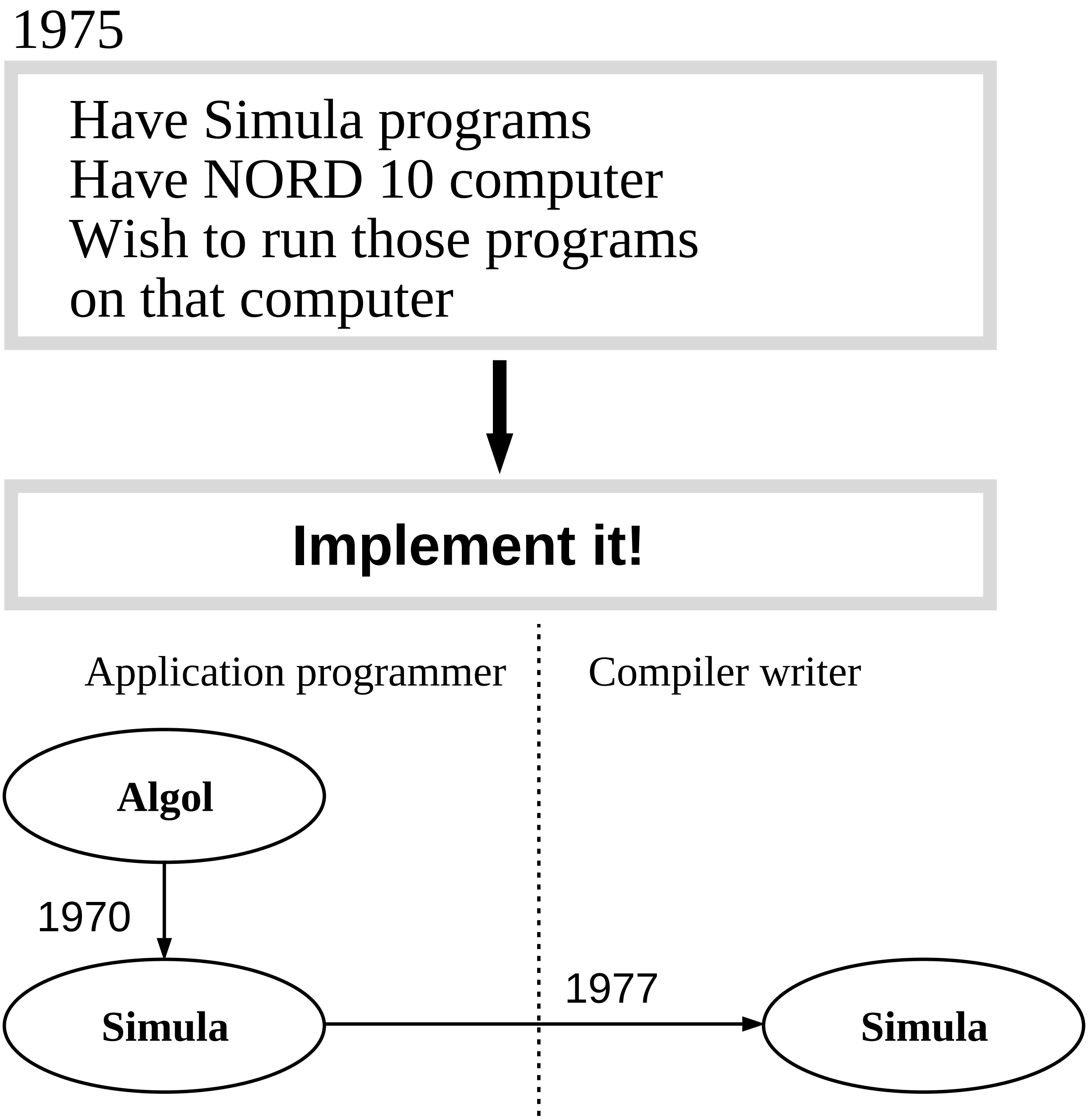
1978:

- Music editor − enter notes, have them transposed and printed out in a format suitable for performing
  - − Input device: Tandberg screen
  - − Output device: DECwriter



- This phase ends when more resources are needed for Simula implementation maintenance, and existing applications move into a maintenance phase too
  - − Documentation is fun!
  - − Maintenance is fun, especially with proper documentation!

Implementation Phase



1975

Have Simula programs
Have NORD 10 computer
Wish to run those programs
on that computer

Implement it!

Application programmer

Compiler writer

1977

Norsk Data AS

A hardware vendor looking for a Simula implementation on their NORD 10

TPH Data AS

A software vendor that "happens to have" a Simula implementation on NORD 10

= Perfect match!

(Matchmaker: Norwegian Computing Center)

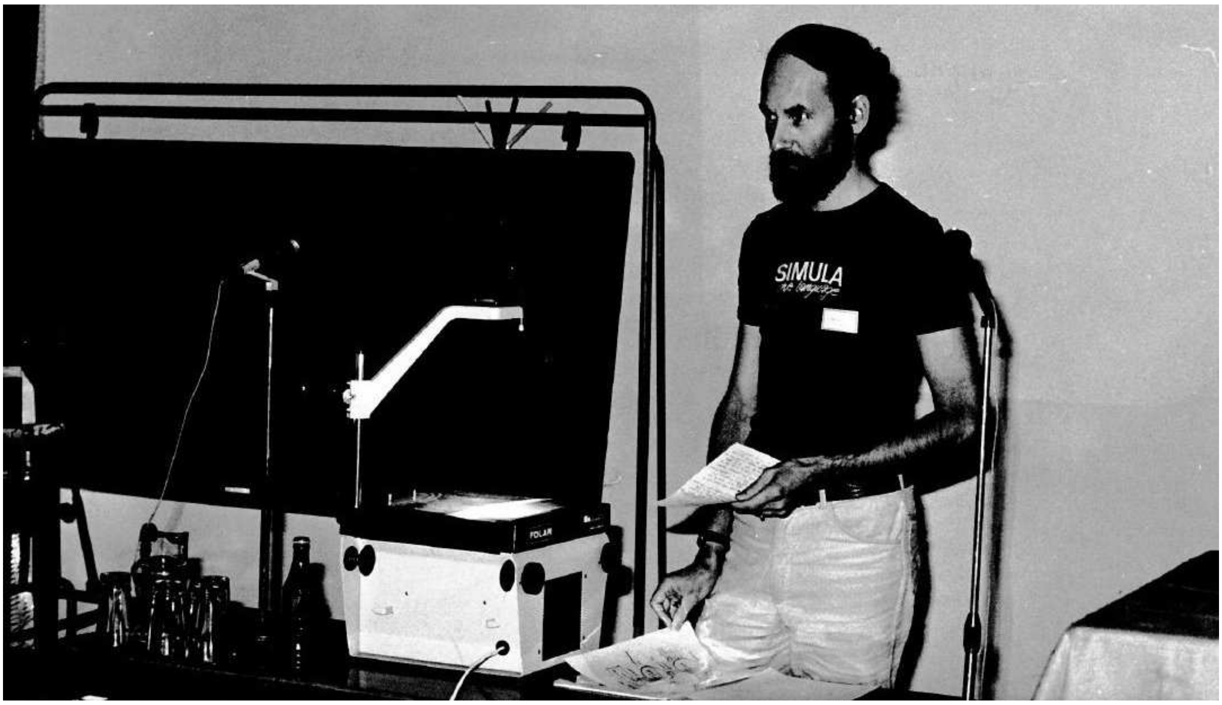
1985

Lectured at a UNIX conference in Copenhagen:

Simula versus C

− The emphasis was mostly on the RISKS of using C, which has a lot of freedom, and must therefore be used carefully and responsibly

− I did not have a lot of good to say about C at the time, having just learnt it and barely used it



1985

Lectured at the same event:

Bjarne Stroustrup:

C++

− Nevertheless, this language prevailed, and later Bjarne told us perhaps why:

1992

Lecture at the 25 years Simula anniversary:

Bjarne Stroustrup:

C++:

Simula in wolf’s clothing?

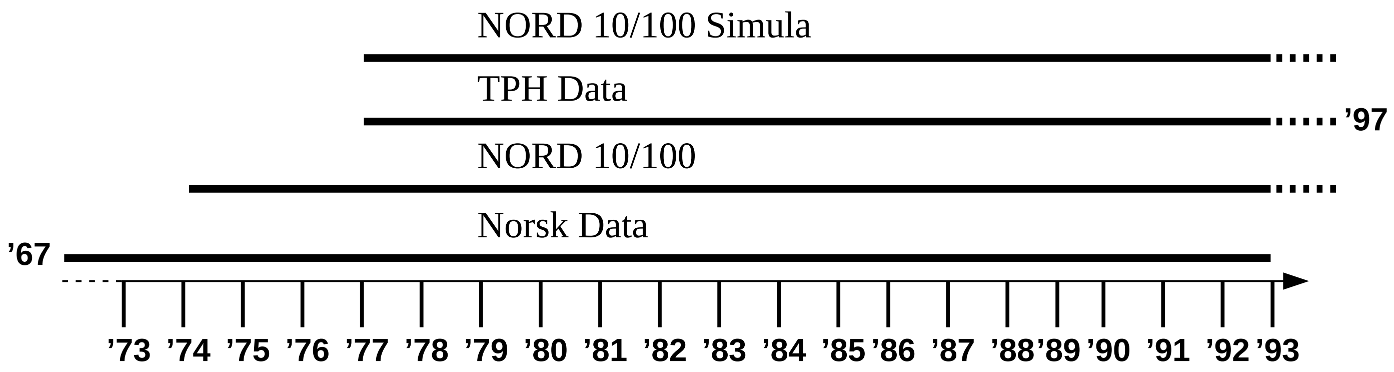
I’m not bad, I’m just drawn that way.
− Jessica Rabbit

So C++ might be ok, it is just the clothing that’s wrong?

1993

This phase ends when Norsk Data closes down, partly for having focused on Sintran rather than on UNIX

(there may be many other reasons for the demise of Norsk Data, but that’s another story)



Leaving the Right Track

1984

- Leaving the Right Track for the Fast Track
  − becoming a C user

In preparation for this:

- Learnt and taught C

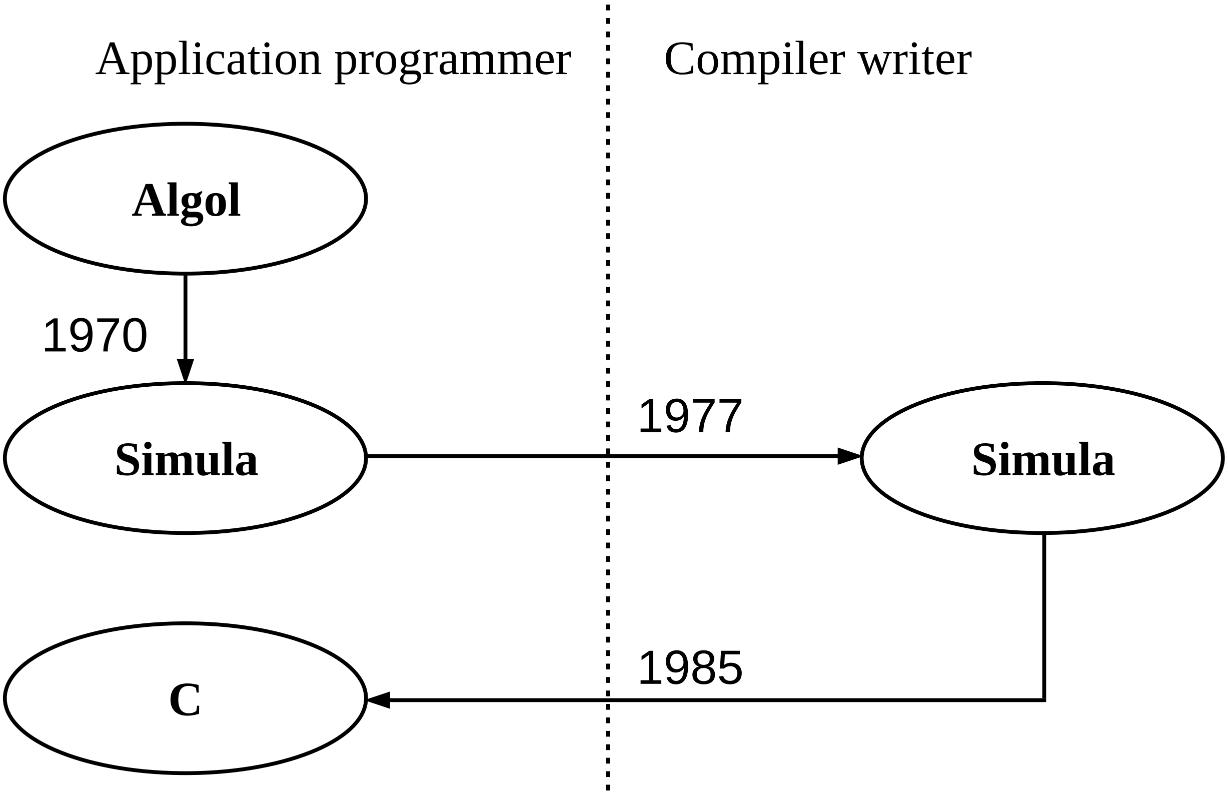
  Yes, I suppose that has to be learnt, too
  − Kristen Nygaard

− To learn a topic fast, teach it
− To learn it thoroughly, use it

Leaving the Right Track

1985

- Wrote UNIX device drivers
  − i.e. operating system components
- Wrote special purpose applications for individual customers
- All in C

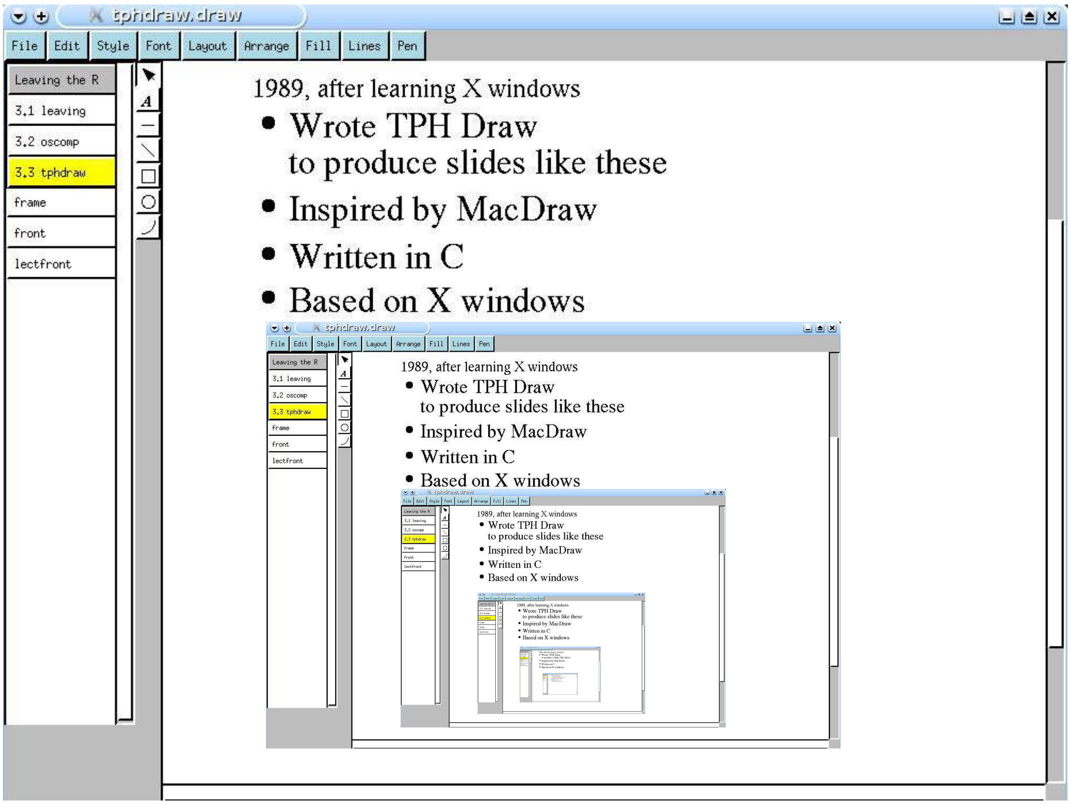


Application programmer

Compiler writer

1989, after learning X windows

- Wrote TPH Draw
  to produce slides like these
- Inspired by MacDraw
- Written in C
- Based on X windows



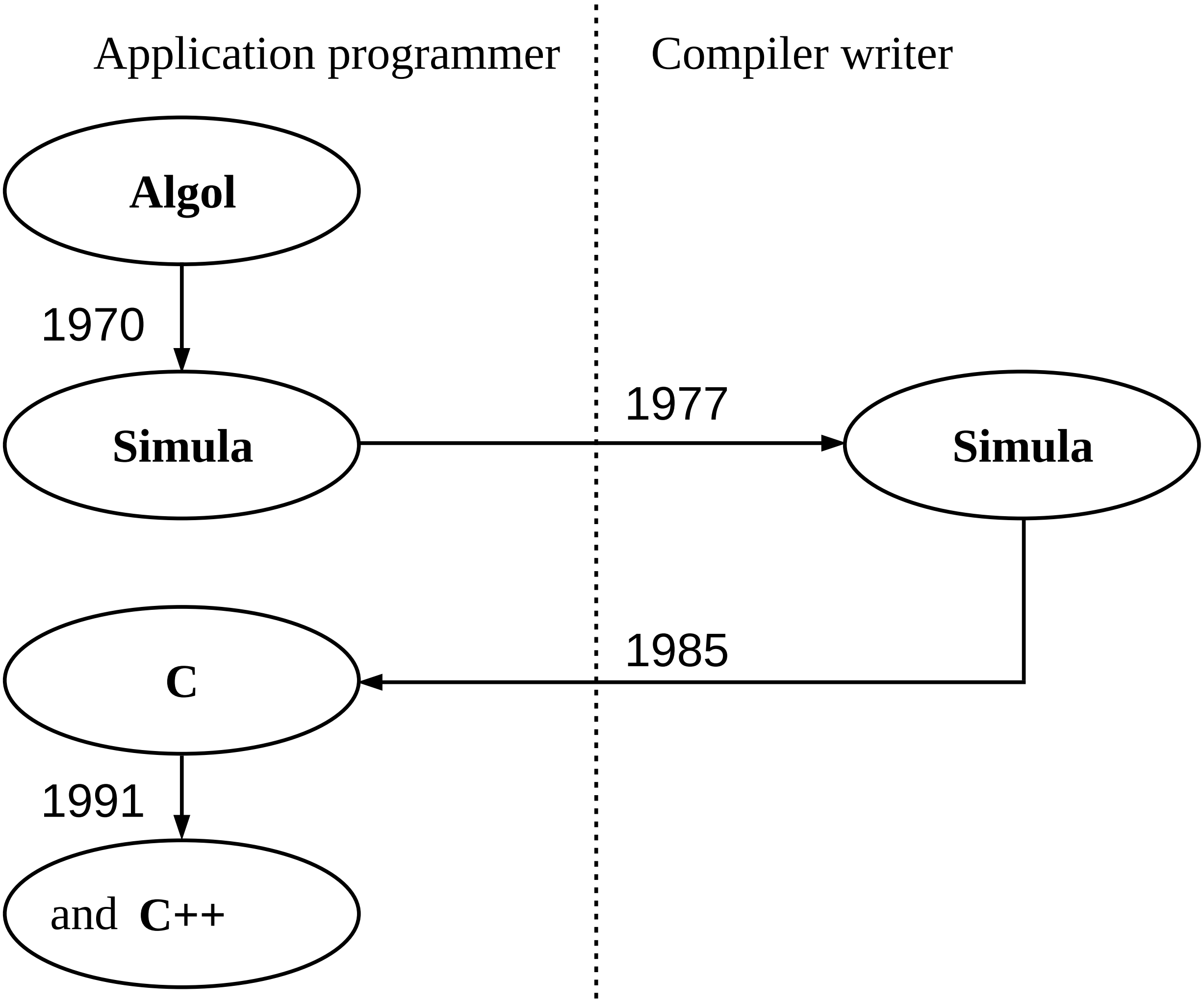
1989, after learning X windows

- Wrote TPH Draw
  to produce slides like these
- Inspired by MacDraw
- Written in C
- Based on X windows

− but I did fake OO in C, so never really dropped the principles of object orientation

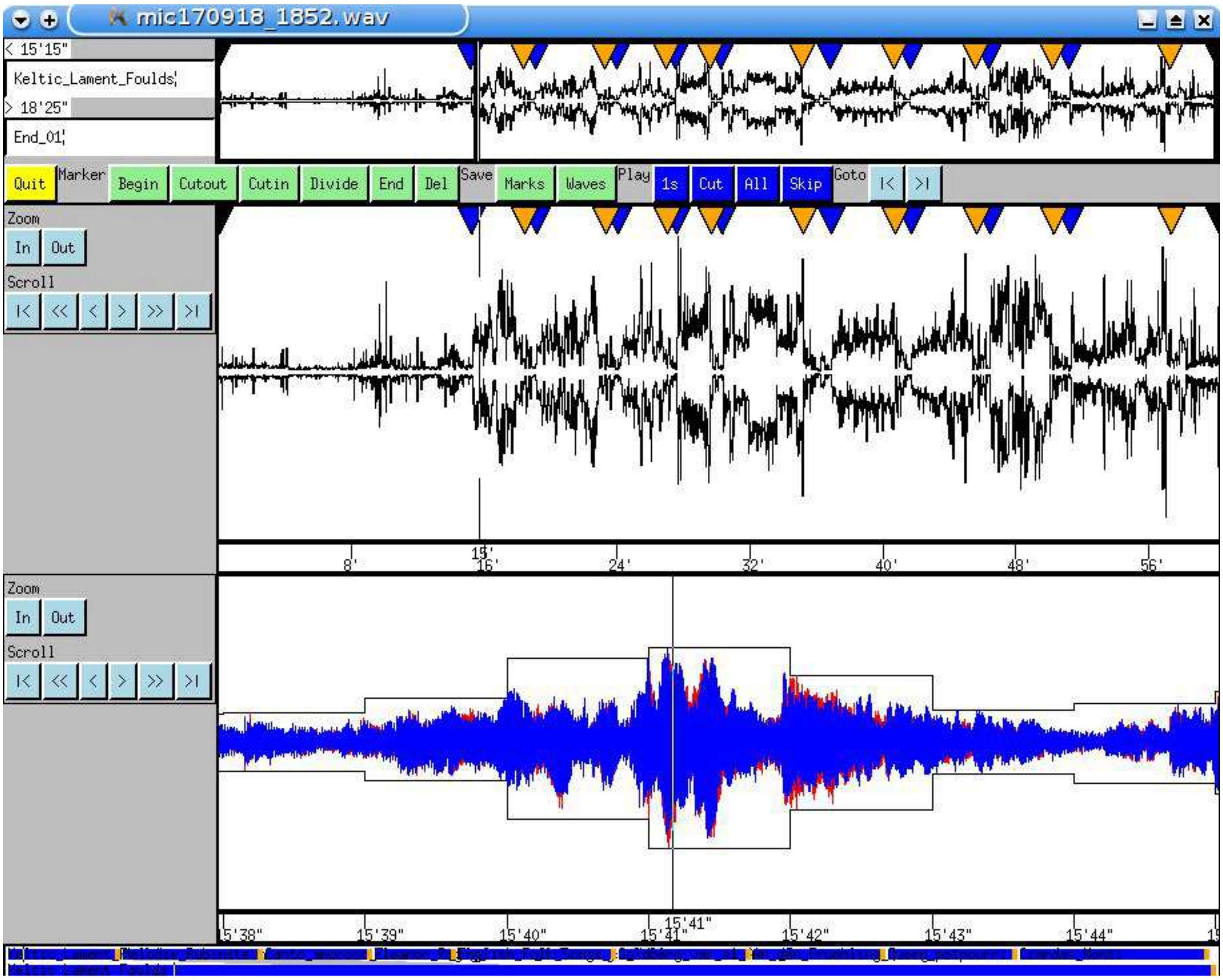
1991

- Learnt and taught C++



Object Oriented Again 4.2

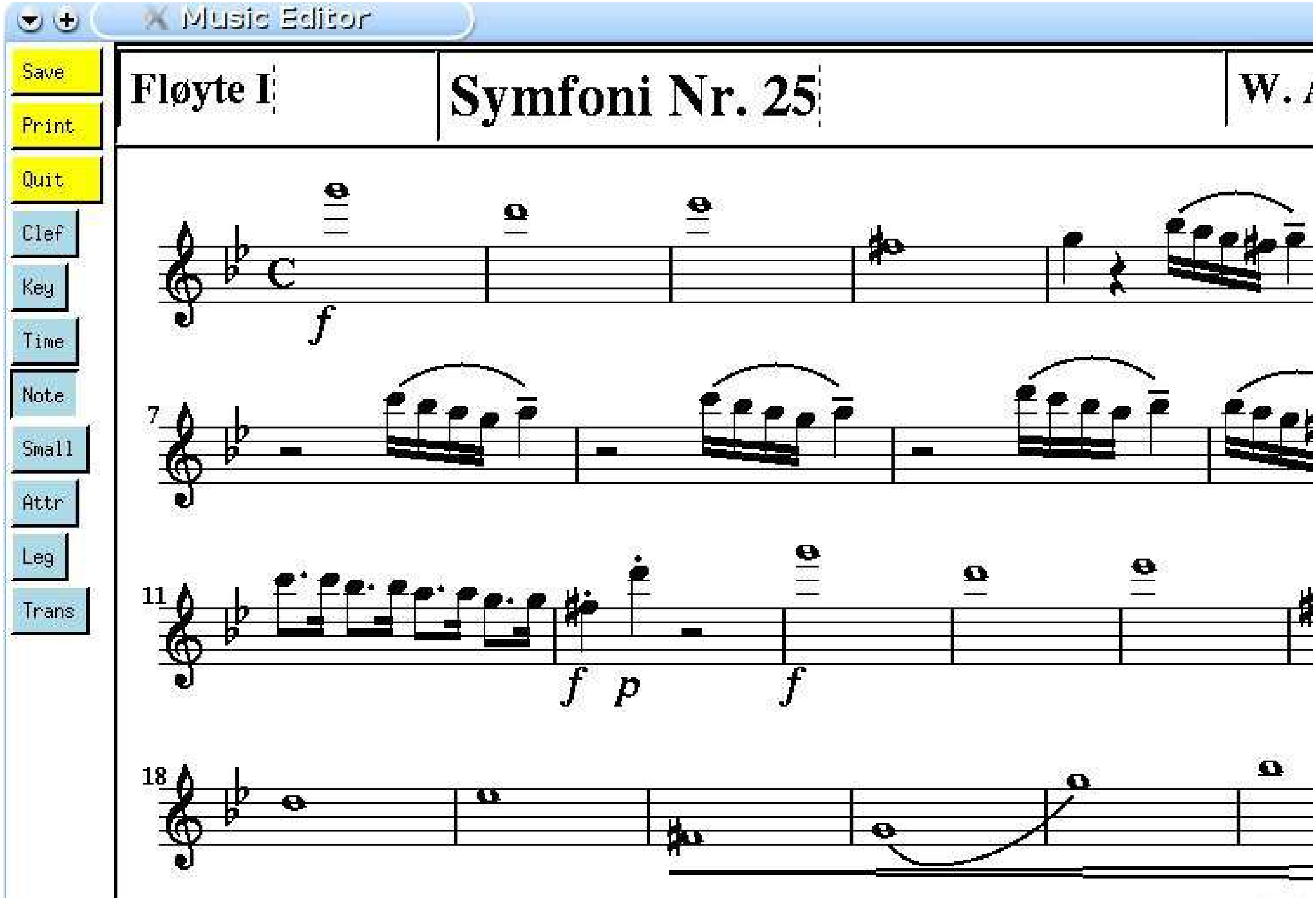
1998
A long awaited successor to the "music" program from 1976:



wedit can divide an audio recording into clips, which in turn can be written to .wav files, each containing one or more clips.
Based on: C++, a viewtools package, and X windows

2001
tdraw, the C++ version of TPH Draw
2002
A successor to the music editor from 1978, which could only show one line of score at a time on the Tandberg text only screen:

medit



Object Oriented Again

1997

- Learnt Perl

> - *a colleague whispered in my ear that it would be a good idea to learn this language*
> - *this was also how I "discovered" Simula in 1970*

2000 −

- Wrote numerous customer applications

+ Documentation and source code can be handed over to customer, often as a single file

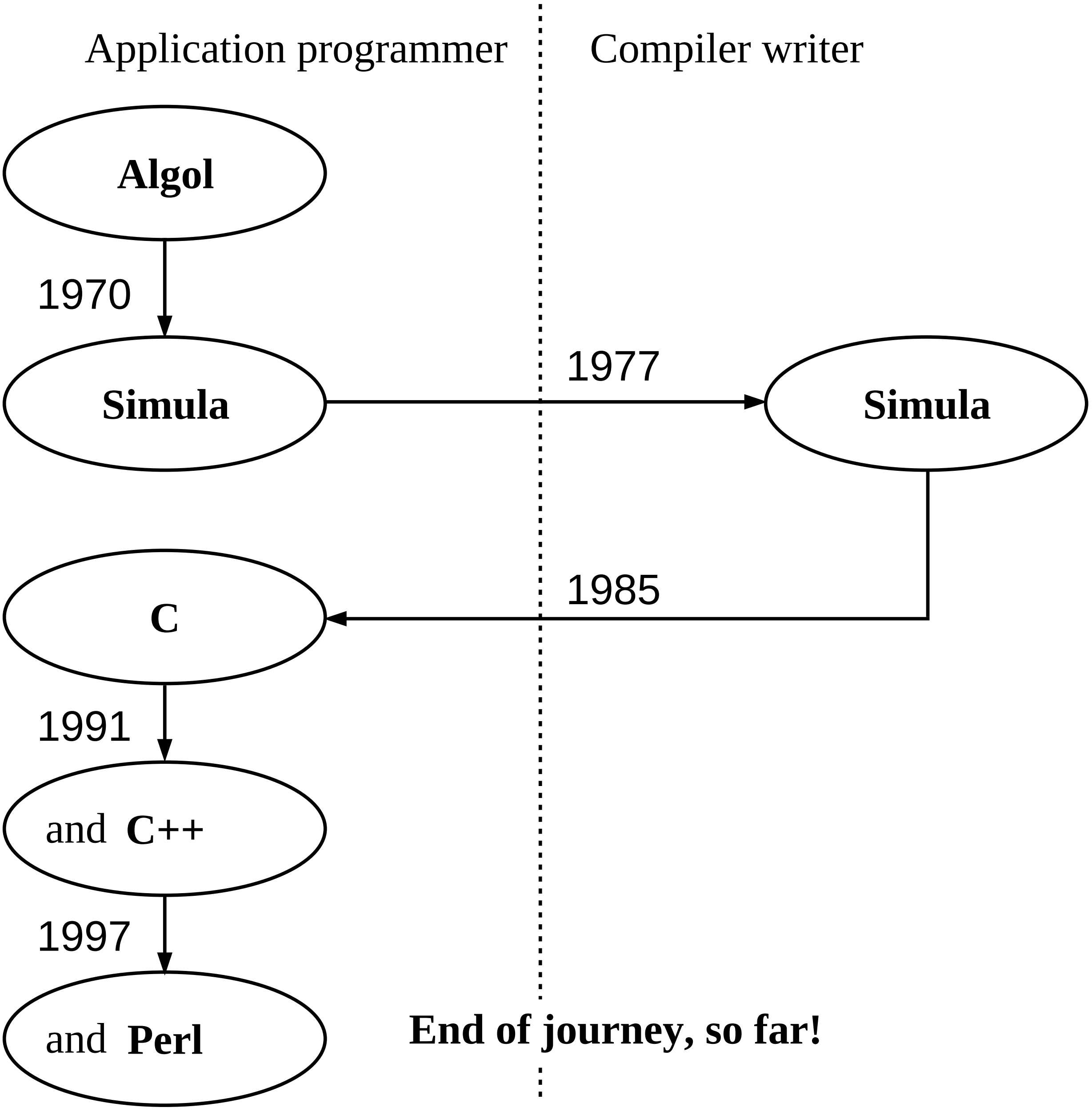
− Forget about hiding your algorithms, but who needs to do that in these open source times?

+ Similar to C, you get a lot of freedom to omit declarations, requalify objects, and other scary things

− With freedom comes responsibility

+ Supports object oriented programming

− Perl is interpreted, not compiled, so slower than C



Application programmer

Compiler writer

Algol

1970

Simula

1977

Simula

1985

C

1991

and **C++**

1997

and **Perl**

End of journey, so far!

Conclusion

Seen from my own individual observation point and practice

- Simula lost as a practical language because:
  I need to deliver code that also my users can compile by default installed compilers
- Simula usually priced separately, not bundled, and certainly not preinstalled
- That is why I left the "pure path" of Simula

Brett Hallett in his Simula Newsletter article of August 1980
"Why did SIMULA miss the boat?" observes

- No complete definition of the language, the Common Base being just an appendix to the base language ALGOL
- The promotion of Simula as a simulation language has served to obscure Simula's more generally useful features and possibilities − the majority of programmers need a general language, not a simulation language

Geoff Collyer in a reply of November 1980 adds

- Simula seems to be large and sometimes awkward
  − Not very orthogonal
  − Special cases and restrictions
  − The language description is quite long

My own early objections to C and C++ were overcome by improvements to both:

- More checks at compile time to avoid program crashes at run time
- Better debugging tools − important because, as every programmer knows, there will always be bugs

Perl in most cases can replace Shell scripts and do a better job because:

- All the work can be done in a single process
- Code needed by several Perl scripts can be put in shared modules
- Even Perl allows object oriented programming if you want it and stay disciplined

− and so I use C, C++, and Perl these days

Still, Simula won in its own way:

- Simula was a major influence for all object oriented languages that followed
- The object oriented way of thinking prevails

− and a celebration is therefore justified!