15.1 History

Verilog is a French company which specialises in the fabrication of tools for software engineering. It produces tools covering all phases of the development of software (analysis, design, code generation, tests, simulation, documentation). These tools are available on different types of computers and different operating systems.

As with many manufacturers of software products, it handles several products. Each of them:

- Has at least one new version each year;
- Runs on several versions of operating systems on several machines;
- Must be up-to-date with respect to the current standards (MOTIF Tool Talk etc.);
- Must be coupled with other products (other tools for PAO, development environments, version and configuration managers, etc.) which themselves are released regularly in new versions and which provide new facilities which have to be taken into account.

These projects must, of course, be open to enable it to allow to be adapted to the clients’ particular needs.

We arrive thus at a combination of the form:

\[ P \times OS \times STD \times CT \times SA \times V \]

- \( P \): products
- \( OS \): operating systems
- \( STD \): standards
- \( CT \): coupled tools
- \( SA \): specific adaptations
- \( V \): versions

which quickly become impossible to handle

In 1988, Verilog decided to implement a reuse policy in order to manage this combination.
The goal aimed at by the introduction of reuse was to pass to a combination of the type:

\[(P + OS + STD + CT + SA) \times V\]

which is much simpler to manage. A product then became an assembly of components amongst which are found components for standard items, couplings with other tools and handles allowing specific modifications.

In 1988 the object oriented techniques were less developed than today. The analysis and design methods for object orientation were at am early stage. The principle languages which were available were:

- Smalltalk, only in interpreted mode;
- C++ at that time only with single inheritance;
- Eiffel 1.0;
- Objective C;
- Simula;
- Ada.

A comparison of these different languages highlighted Eiffel for several reasons:

- The compiled mode was more efficient than Smalltalk;
- Language typing which made it more reliable than Smalltalk;
- Multiple inheritance was considered important for less restricted reuse (single inheritance in Smalltalk and C++);
- The existence of generics (absent in C++ at that time);
- The simplicity, the clarity and power of the language;
- The concept of programming by contract.

A Pilot Project, realised by a single person in 1987, with the Eiffel compiler had allowed us to verify the qualities of language.

### 15.2 First Experience with Object Orientation

Starting in 1988, a new pilot project employing two persons was started. It was immediately necessary to face up to a set of problems due to the absence of configuration management. The two developers shared the same classes. While one of them made changes, there was recompilation for the other because the Eiffel compiler controls consistency between classes. To avoid this problem it was necessary to duplicate the classes for each user. The solution was judged insufficient because with 100 developers, there was no question of copying all sources 100 times. The use of a version manager such as SCCS also appeared to be proscribed because, before compilation, it was necessary to remove all sources which also used a lot of space.

The providers of the Eiffel compiler did not provide their compiler with a management or configuration manager. Verilog then decided to develop its