A Biography by the Books

Peter FELECAN

June 15, 2009

Contents

1 Introduction

Reading is a formative passion. A passion which never let me down from the earliest times. What's more natural then, than to write a biography as an essay geared by the books[1] which marked my professional career? Somewhere else, I will try this exercise for my personal endeavors.

2 Background

Trained as a software engineer, I have an MS in Computer Science. The magic of mathematics, the internal organization of systems and their relationship with the environment, drawn me early to computers, writing my first program at the age of 14.

3 Career Trail

My professional career started in France — I'm a French national with Hungarian origins — at the beginning of the 80s.

[1] All the cited books have a link toward a more general bibliography.

The first stints were as a system programmer in what will became my predominant domain of activity: research and development for software publishing houses. Designing and implementing software tools for development teams creating shrink-wrapped business oriented products, made the foundation of my professional life.

Later, I took up the management of a methodology, procedures and tools division for a financial software publishing house. Managing, guiding and tutoring engineers engaged in the support of the life cycle for a large offering for small and medium companies.

After dwelling in information retrieval, the energy industry and mobile telecommunication as a systems architect, the natural evolution brought me at the head of the research and development division for a multi-cultural company providing products for content delivery networks.

Since the end of the Internet frenzy, I managed the research and development division for various software publishers in the financial and document production domains.

Despite the vertical variety of application domains, my specialization is horizontal: software engineering.

4 Technical Profile

If there is something which could concisely characterize my technical profile is Open Systems and Free Software.

Since the beginning of my professional activity, I had the luck to work with numerous Unix implementations, from BSD and Xenix, to Solaris and AIX, passing through HP-UX and Linux. The level of acquaintance gained from this is quite intimate, in part thanks to many iterations in reading the enlightening Design of The Unix Operating System [2] by Maurice J. Bach.

Given the heterogeneity of the IT environment, I had to wander, quite often, through the arcades of VMS, MS-DOS and their Windows descendants, in all their splendors.

The need to interconnect these systems made me familiar with the network protocols, on many layers, as TCP/IP and HTTP, just to name a few.

Programming languages are part of my more general interest in linguistics. Even wrote some, used internally in different projects, perusing `lex/flex` and `yacc/bison`. In those periods, the *Dragon Book*[1] is a mandatory (re)reading.

As many of my colleagues starting their professional career at the beginning of the 80s, I was introduced to the universally terse but generally efficient C language by copiously reading the *White Book*[6] by Brian Kernighan and Dennis Ritchie. Well, it changed my practice from assembler, FORTRAN or COBOL.

The following years and many lines of C code, were used for a better understanding of operating systems, by Rochkind's *Advanced Unix Programming*[11] , and software tools by Plaugher's classic[10] . The traps and falls of this period were avoided thanks to Andrew Koening[8] .

I discovered Smalltalk and the object oriented paradigm around 1985. Reading the seminal book by Brad Cox, Object Oriented Programming, An evolutionary Approach[5] convinced me and my colleagues to buy into software integrated components and a C++ license from AT&T, consequently meeting Bjarne Stroustroup, unfortunately only through his books[13] [14] ...

The object oriented paradigm was a companion through many years of, not only coding, but also designing. Yourdon and Coad methods where my friends[3] [4] . Later, UML gently replaced their models[12] . One of the way of exercising my knowledge in UML was to write a full design tool in Java for one of my customers in the energy industry — this was in 1995, well before Together or Poseidon, but slightly after Rational Rose.

Depending on the projects in which I was involved, I made forays in other languages: Lisp, PostScript, TEX[7] and LATEX[9] , scripting in sh, ksh, bash and Perl, etc.

Almost everything in my activities needs, at a moment or another, configuration management. Many systems vaunting their prowess entered in my usage, from SCCS to CVS, passing through NSE, with it's magnificent Translucent File System, and SourceSafe... I always come back to CVS. Well, `subversion` has some nice features and integration with other software engineering tools which brought me to using it in some projects.

I'm an enthusiastic proponent of Emacs, although vi it's an old buddy.

Working in the information retrieval domain brought me close to the data management arena, where I fought along numerous incarnations of RDBMS.

On the methodology side, I implemented in the research and development organizations that I managed, various life cycle approaches, such as ISO 12207, RUP and the CMM.

As a last note, I'm a long date and still fervent reader of Knuth's writings. Even the religious ones — they have so nice illustrations.

5 Types of mission

The types of mission where I feel efficient, are:

- Operational management and team building.
- Technology strategy and audit for emerging software organizations in relation with investment funds.
- Product management.
- Software publishing.
- Architecture of information systems.
- Object Oriented analysis, design and implementation.
- Fundamental software engineering.
- Porting and packaging of complex projects.
- Audit, optimization and security for system software.
- Training and coaching.

6 Languages

Born in a multi-cultural family, having lived in different countries and working for international companies, gave me the opportunity to master 4 European languages.

My interest in linguistics, brought me near ancient cultures and their magnificent languages.

7 Professional Organizations

A long date member of Association for Computing Machinery.

8 More information

If you want to dig in the details, you can ask for my curriculum vitae and, if it's important for you, even a photo will be sent... Just use the contact information available on this site.

As you can see in the menus situated at the top and the bottom of this page, there is also a French version of this document.

9 Bibliography

References

[1] Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman. *Compilers: Principles, Techniques and Tools*. World Student Series. Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

Also known as the *Dragon Book*.

[2] Maurice J. Bach. *The Design of the Unix Operating System*. Software Series. Prentice Hall, Prentice Hall PTR, Englewood Cliffs, New Jersey, 07632, 1986.

For every aspiring system engineer.

[3] Peter Coad and Edward Yourdon. *Object-Oriented Analysis*. Prentice-Hall, Prentice Hall PTR, Englewood Cliffs, New Jersey, 07632, second edition, 1991.

.

[4] Peter Coad and Edward Yourdon. *Object-Oriented Design*. Prentice-Hall, Prentice Hall PTR, Englewood Cliffs, New Jersey, 07632, 1991.

Je ne suis pas sûr que la traduction française porte sur le bon ouvrage.

[5] Brad J. Cox. *Object Oriented Programming, An Evolutionary Approach*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

Il n'y a pas de traduction française connue.

[6] Brian W. Kernighan and Dennis M. Ritchie. *The C Programming Language*. Software Series. Prentice-Hall, Prentice Hall PTR, Englewood Cliffs, New Jersey, 07632, 1978.

A reference for the language and for writing a book defining one.

[7] Donald E. Knuth. *The TeXbook*. Professional. Addison-Wesley Publishing Company, Reading, Massachusetts, 1984.

Il n'y a pas de traduction française.

[8] Andrew Koening. *C Traps and Pitfalls*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.

A very good explaination of variadic functions and of the printf family.

[9] Leslie Lamport. *LaTeX: A Document Preparation System*. Professional. Addison-Wesley Publishing Company, Reading, Massachusetts, second edition, 1994.

À ma connaissance il n'y a pas de traduction française.

[10] P. J. Plauger and Brian W. Kernighan. *Software Tools*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1976.

Learning by mimetism.

[11] Marc J. Rochkind. *Advanced Unix Programming*. Software Series. Prentice-Hall, Prentice Hall PTR, Englewood Cliffs, New Jersey, 07632, 1985.

For every aspiring system programmer.

[12] James Rumbaugh, Ivar Jacobson, and Grady Booch. *The Unified Modeling Language Reference Manual*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1999.

Pour la traduction française il s'agit de la version 2.0 d'UML.

[13] Bjarne Stroustrup. *The C++ Programming Language*. Computer Science. Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

.

[14] Bjarne Stroustrup and Margaret A. Ellis. *The Annotated C++ Reference Manual*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1990.

Il n'y a pas de traduction française connue.