

# Christophe Delord

<b>Personal data</b>	
Christophe Delord	<b>Software Engineer</b> - R&D - Open-Source Age: 51 year old - born in 330.5 PPM contact: <a href="mailto:cdelord.fr">cdelord.fr</a> - <a href="#">cdsoft.codeberg.page</a> - <a href="#">codeberg/cdsoft</a> - <a href="#">github/CDSOft</a> - <a href="#">LinkedIn</a>
<b>Experience</b>	
Computer science	<b>Computer Science Engineer</b> <b>Post Graduate Degree in Artificial Intelligence</b> <b>ENSEEIH</b> 27 year experience (artificial intelligence, natural language processing, genetic algorithms, specification, design, unit testing, integration, validation, embedded computers, avionics, automotive, R&D, Open-Source...)
<b>Technical Skills</b>	
Programming	<ul style="list-style-type: none"><li>functional (<b>Haskell</b>, CaML, LISP),</li><li>logic (<b>Prolog</b>),</li><li>imperative (<b>C</b>, Ada, Pascal, <b>Python</b>, <b>Lua</b>),</li><li>object (Java, <b>C++</b>, Eiffel, Pascal, <b>Python</b>),</li><li>mathematics (FORTRAN, Xcas),</li><li>low level (Assembleur (80x86, 680x0, SHARC, PowerPC, PIC32), PL/M)</li><li>Web (HTML, Javascript),</li><li>script (bash, Perl, <b>Python</b>, <b>Lua</b>, TCL)</li></ul>
Methods	formal specification (event-B, Rodin), artificial intelligence,
Safety-critical standards	DO-178B (avionics), ISO 26262 (automotive)
Architecture	Intel (80x86), Motorola (680x0), VHDL, SHARC (2106x), PowerPC (MPC5554), Microchip (PIC32)
Operating Systems	UNIX, GNU/Linux (Debian, Fedora, Shell, Perl, Python, Tcl/Tk, C, ...)
Version control	Git
Publishing	LaTeX, reStructuredText, <b>Markdown</b> , <b>Pandoc</b>
Open Source	<a href="#">LuaX</a> (programming language, libraries), <a href="#">bang</a> (build system), <a href="#">ypp</a> , <a href="#">panda</a> (text preprocessing), <a href="#">tpg</a> , <a href="#">sp</a> (parser generators)...
<b>Patents</b>	
Dec. 20, 2019	Method and system for handling blind sectors of scanning layers of redundant sensors in a vehicle. See <a href="#">patents.google.com</a> or <a href="#">patents.justia.com</a>
<b>Professional Experience</b>	
Feb. 2017 - ...	<b>EasyMile. Toulouse.</b> <ul style="list-style-type: none"><li>Real-time embedded software (C, Lua, Ethernet, CAN)</li><li>Sensor (LiDAR) and environment (vehicle and moving obstacles) simulation (Haskell, Lua, Python, Ethernet, CAN, Linux)</li></ul>
Studies	<b>Sopra</b> <ul style="list-style-type: none"><li>Evaluation of formal methods (<a href="#">event-B</a>, <a href="#">Rodin</a>)</li><li>Usage of functional languages (Haskell, OCaml, F#) to model real-time embedded systems</li><li>Artificial intelligence applied to automatic unit test generation</li></ul>
Aug. 2015 - Jan. 2017	<b>Sopra for Airbus, Simulation. Toulouse.</b> <ul style="list-style-type: none"><li>Real-time simulation for flight computers (Simics, Power PC, Linux, AFDX)</li></ul>
Sept. 2014 - Jan. 2017	<b>Sopra for Airbus, Flight test. Toulouse.</b> <ul style="list-style-type: none"><li>A330 Neo flight tests optimisation. Study on the process and tools for the aircraft instrumentation.</li><li>Wi-Fi network optimisation of the A350 flight test installation.</li><li>Real-time Linux OS</li><li>Study of a real-time physical parameter acquisition modules (Microchip PIC32 microcontroller, clock synchronisation, C).</li></ul>
Sept. 2014	<b>Sopra for Thales Avionics. Toulouse.</b> Qualified ARINC 665 load generator <ul style="list-style-type: none"><li>Design and code (C)</li><li>Evolution</li></ul>
Jul. 2014 - Aug. 2014	<b>Sopra Group for Thales Optronique. Élancourt.</b> Real-time modular test bench (design, code, tests) <ul style="list-style-type: none"><li>real-time kernel in C++ (Windows and RTX)</li><li>modular and configurable by Python scripts</li></ul> (Windows, RTX, C++, embedded Python interpreter)
June 2014 - June 2014	<b>Sopra Group for Liebherr-Aerospace. Toulouse</b> Specification, design and code manual verification (KC 390, SW-LR)
June 2014 - June 2014	<b>Sopra Group for Liebherr-Aerospace. Toulouse</b> Unit testing (C, RTRT, SCADE, automatic test generation in Python, RTRT)

Mar. 2014 - May 2014	<b>Sopra Group for Airbus. Toulouse.</b> Flight Control SECondary Computer test (A350) (CMM level 3, DO-178B level A, Sharc Assembly, integration, validation, JScript, Perl, Python, C).
Feb. 2014 - Feb. 2014	<b>Sopra Spain for Fermox. Valencia, Spain.</b> Study for a VoIP intercom with Sopra Valencia (VoIP, Microchip IC32 microcontroller, real-time, C).
Oct. 2013 - Mar. 2014	<b>Sopra Group for Thales Avionics. Toulouse</b> Qualified ARINC 665 load generator <ul style="list-style-type: none"> <li>• Design and code (C)</li> <li>• Generic data formatting system (symbolic description of data formats and their relationships, automatic formatting and generation).</li> </ul>
Sept. 2012 - Nov. 2013	<b>Sopra Group for Thales Optronique. Élancourt.</b> Real-time modular test bench (design, code, tests) <ul style="list-style-type: none"> <li>• real-time kernel in C++ (Windows and RTX)</li> <li>• modular and configurable by Python scripts (Windows, RTX, C++, embedded Python interpreter)</li> </ul>
Apr. 2012 - Oct. 2012	<b>Sopra Group for Liebherr-Aerospace. Toulouse</b> Onboard Maintenance System (OMS) simulator (DO-178B niveau B): <ul style="list-style-type: none"> <li>• design, code and test of an OMS</li> <li>• graphic user interface to drive the BITE function of a LRU</li> <li>• ARINC 604 protocol over an ARINC 429 link</li> <li>• Python scriptable test environment</li> <li>• ARINC 604 protocol test</li> <li>• BITE LRU simulation (to test and validate the test environment)</li> <li>• Sphinx documentation project, automatic documentation generation (design, traceability matrices, test reports)</li> </ul> (Python, C, reStructuredText / Sphinx documentation, SVN, automatic documentation generation)
Jan. 2011 - Sept. 2012	<b>Sopra Group for Airbus. Toulouse.</b> Flight Control SECondary Computer (A350) (CMM level 3, DO-178B level A, Sharc Assembly, unit testing, integration, validation, JScript, Perl, Python, C, DSP simulation for performance and robustness validation). Microprocessor simulation (time and stack usage measure, Python, Optimized graph searched)
Jun. 2008 - Jan. 2011	<b>Sopra Group for Thales Avionics. Toulouse/Paris.</b> A320 flight control secondary computer redesign (DO-178B level A and D, MPC5554, Assembly, C and ADA, Specifications, Design, Code).
Mar. 2007 - Oct. 2008	<b>Sopra Group for Airbus. Toulouse.</b> Specification of an embedded Onboard/Ground communication system for Airbus (Wifi, GSM, VPN, ...).
Jan. 2007 - Feb. 2007	<b>Sopra Group for Airbus. Toulouse.</b> Unit testing for an Airbus embedded calculator (A400M), training of a testing team in India.
Jan. 2007 - Jul. 2007	<b>Sopra Group. Toulouse.</b> TOPCASED: Toolkit in OPen-source for Critical Application and SystEms Development, Safety study. Contribution to the   AESE conference for the centenary of ENSEEIHT.
Nov. 2006 - Dec. 2006	<b>Sopra Group for Airbus. Toulouse.</b> Flight Warning Computer (A400M), coding rules and unit testing (DO-178B, Level B).
Mar. 2002 - Oct. 2006	<b>Sopra Group for Airbus. Toulouse.</b> Flight Control SECondary Computer (A380) (CMM level 3, DO-178B level A, Sharc Assembly, unit testing, integration, validation, TCL, Perl, Python, C, DSP simulation for performance and robustness validation). Microprocessor simulation (time and stack usage measure, Python, Optimized graph searched)
Oct. 2001 - Mar. 2002	<b>Sopra Group for Airbus. Toulouse.</b> Flight Control Primary Computer (A330/340) Validation (DO-178B, Level A, Intel Assembly).
May 2001 - Oct. 2001	<b>Sopra Group for Airbus. Toulouse.</b> Update of the Flight Warning System (A340) for a certification, update of the software life cycle (DO-178, Intel Assembly, PL/M, ADA).
Jul. 1999 - May 2001	<b>Sopra Group for Pierre Fabre Laboratories. Castres.</b> Communication between data bases and distant PC (Unix, Shell, Perl, C).
Oct. 1998 - Jul. 1999	<b>Sopra Group for CNRS. Labège.</b> Correction and evolution of the "Accounting and Financial Management" application of the CNRS.
1997 - 1998	<b>ENSEEIH-IRIT. Toulouse.</b> DEA training period and ENSEEIHT 3rd year: Modeling of the cognitive process of dialogue (Prolog, Speech Acts, ...).

## Personal Projects

<b>Actor model</b>	Designing safety critical real-time systems with <a href="#">actors</a> in <a href="#">LuaX</a> and C <ul style="list-style-type: none"> <li>• high level DSL in LuaX with expressive dynamic types</li> <li>• low level C generated code with strong static types</li> </ul>
<b>Modeling and simulation</b>	Usage of functional programming ( <a href="#">Haskell</a> ) to model and simulate critical real-time systems <ul style="list-style-type: none"> <li>• strong static typing → type system proofs replace some integration activities</li> <li>• pure functional programming → no side effect, determinism, testability</li> </ul>
<a href="#">Belenos</a>	<b>Model checker in C</b> A basic model checker that simulates the execution of a system model written in C.
<a href="#">BonaLuna, LuaX</a>	<b>Lua extension</b> A small, standalone and extendable Lua interpreter providing portable scripting features for Windows, MacOS and GNU/Linux.
<a href="#">bang</a>	<b>Ninja file generator scriptable in LuaX</b> Combine the speed of Ninja and the expressiveness of LuaX to write efficient build systems.
<a href="#">PP, ABP, Panda, UPP, ypp</a>	<b>Text preprocessor</b> designed for <a href="#">Pandoc</a> , Markdown and reStructuredText written in <a href="#">Haskell</a> and <a href="#">Lua</a> <ul style="list-style-type: none"> <li>• text macros</li> <li>• user defined macros</li> <li>• diagrams</li> <li>• scripts</li> <li>• <a href="#">literate programming</a></li> </ul>
<a href="#">Functional specifications</a>	<b>Formal methods</b> Functional languages (Haskell) used to formally describe and verify a system
<a href="#">PopF</a>	<b>Unsolicited Emails Filtering</b> Statistical filter, POP3 Proxy
<a href="#">PyLog</a>	<b>First order logic and PROLOG in Python</b> First order terms and variables, PROLOG inference engine, PROLOG to Python translator
<a href="#">TPG</a>	<b>Toy Parser Generator</b> A lexical and syntactic parser generator for Python (Recursive descendant parser, Attributed grammars, Abstract syntax tree building).
<a href="#">SP</a>	<b>Simple Parser</b> Another lexical and syntactic parser generator for Python (Recursive descendant parser, Backtracking, Functional Programming, Abstract syntax tree building).

## Student Projects

1997 - 1998	<b>ENSEEIH - 3rd year Student</b> ENSEEIH/DEA training period (human dialogue simulation).
1996 - 1997	<b>ENSEEIH - 2nd year Student</b> Compilation of a subset of C-language, execution in a virtual machine (Eiffel, C) Object oriented design and programming (Eiffel) Expert Systems, Predicate Logic (Prolog) Operating systems, client/server (HTTP server) (Unix, C) Hardware (calculator, pipeline, ...) (VHDL)
1995 - 1996	<b>ENSEEIH - 1st year Student</b> Hardware, microprocessor (and biprocessor) design and simulation in C++ (as a personal project) Cryptography (C) Expert Systems (Lisp)
Taxia	<b>Embedded computers in a taxi</b> Event programming, Gui, C++, assembly.
Hardware, simulation	<b>Biprocessor simulation (see 1st year)</b> (C++, HP48), Schip-48 virtual machine and disassembler (C)

## Other Experiences

Summer 1993	Development of a data-base software for pupil registration management
1993 - 1998	Private lessons (Mathematics, Physics, Computer Science)

## Education

1997 - 1998	<b>Post Graduate Degree in Artificial Intelligence</b> ENSEEIH-IRIT, Toulouse
1995 - 1998	<b>Computer Science Engineer (10th)</b> ENSEEIH, Toulouse
1998	<b>Test Of English for International Communication (820/990)</b> Toulouse
1994 - 1995	<b>Two year degree in Mathematics and Physics</b> Paul Sabatier University, Toulouse
1994	<b>Cambridge Examinations (First Certificate in English)</b>

1993 - 1994 | Lycée Pierre de Fermat, Toulouse  
**Preparatory classes**  
Lycée Pierre de Fermat, Toulouse

### Publications

Sep. 1998 | **Christophe Delord. Actes de langage et jeux de dialogue.**  
Human dialogue simulation. ENSEEIHT-IRIT, Toulouse, France

Sep. 1998 | **Christophe Delord. Actes de langage et jeux de dialogue.**  
Introduction of a human dialogue simulation model.  
In Colloque Intelligence Artificielle et Complexité (I.A.C'98), Saint Denis  
University - Paris VIII

### Languages

French | Native Speaker  
English | Intermediate  
German | Working Knowledge