Curriculum Vitæ

Didier Verna
Epita/ Lre
14–16 rue Voltaire,
94276 Le Kremlin-Bicêtre
France

didier@didierverna.net
https://www.didierverna.net/

0000-0002-6315-052X profile/Didier_Verna in/didierverna

Distinctions

Keynote Speaker at <Programming> 2026
(Verna, 2026a, nº 55)

Reviewers Choice Award 2018 from the Art, Science and Engineering of Programming Journal
(Verna, 2018a, nº 3)

Keynote Speaker at ACCU 2014
(Verna, 2014, nº 58)

Best Paper Award 2006 from the European Lisp Workshop
(Verna, 2006a, nº 47)

Currently

Position:	Professor of Computer Science with Habilitation to Direct Research (HDR)
Teaching Establishment:	École pour l'Informatique et les Techniques Avancées (Epita)
Research Establishment:	Epita Research Lab (Lre)

Career

2024 **Qualification for Full Professorship / Tenure-Track Positions (France)**
French National Council of Universities (CNU)

Section: 27 — Computer Science

Jury: Olga Kouchnarenko, Céline Rouveirol

2020 **Habilitation to Direct Research (HDR), Edite of Paris, Sorbonne University, Epita/Lre**

Title: `(dynamic (programming paradigms)) ;; performance and expressivity`

Jury: Manuel Serrano, Inria, Sophia Antipolis, France (reviewer)
Robert Strandh, Bordeaux University, France (reviewer)
Nicolas Neuß, Fau, Erlangen-Nürnberg, Germany (reviewer)
Marco Antoniotti, Milan-Bicocca University, Italy (examiner)
Ralf Möller, Lübeck University, Germany (examiner)
Gérard Assayag, Ircam, Paris, France (examiner).

2000 **Ph.D. from École Nationale Supérieure des Télécommunications de Paris (Enst)**

Title: Virtual Reality and Tele-Operation: Assisting the Operator with Cognitive Modelling of his Intentions

Jury: Philippe Coiffet, Jean-François Richard (reviewers)
Jean-Paul Papin (examiner)
Alain Grumbach (Supervisor)

1994 Engineering Degree from Enst, specialization Computer Science and Networks
1988 French Scientific "Baccalauréat" (C)

Professional Experience

2026 Bias in Peer Review training course, CIHR / SSHRC / NSERC
2025 Intercultural skills training course, doctoral college, Sorbonne University.
2014–… Member of Edite of Paris (Computer Science, Telecommunications, and Electronics Doctoral School))
2005–… Adjunct lecturer at Sorbonne University/Upmc, Enst, Ensta, French National "Beaux Arts" School and Ionis stm. *Cf. details in the "Teaching Activities" section*
1999–… Full Professor of Computer Science at Epita

Note

I work part-time (initially 3/5th, more recently 4/5th). The following numerical data is expressed in absolute value. Teaching hours are accompanied with a full-time equivalent, but the rest (publication density, ecosystem participation, *etc.*) is to be evaluated relatively to my part-time activity.

1 Administrative Responsibilities

2027 TUG (TeX Users Group Conference) local chair
2026–… National coordinator for the Epita academic research initiation module
2026–… Member of the DocEng (ACM SIGWEB Symposium on DOcument Engineering) Steering Committee
2026 Member of the evaluation committee for the program Samuel De Champlain 2026–2027, section *Computer Science and Software and Electrical Engineering*, Quebec research fund for France-Quebec research teams mobility / FRQ – CFQCU
2025 Lre annual plenary meeting co-organizer
2020–… National coordinator for fundamental computer science lectures, *cf.* section "Teaching Activities"
2014 ELS (European Lisp Symposium) local chair
2009–… Involvement in the recruiting process of Epita students (open days, forums, interviews, *etc.*) *About 90h / year*
2008–… Co-founder and president of the European Lisp Symposium (ELS) Steering Commitee
2008 Founder and first organizer of the Performance and Genericity Seminar of the Epita Research Lab (Lre)

2 Teaching Activities

I am the designer of all the lectures I have been giving since 1999. All course material is publicly available on the teaching page[1] of my scientific website, including archived former lectures.

2.1 Currently at Epita

The lectures below are given to more than 600 3rd year undergraduate students every year. I am the national coordinator for these lectures: I design them and give them myself at the Parisian campus (around 400 students in 3 groups) and I supervise teams of 4 to 6 other lecturers who give them, with my course material, at the four other campuses in France: Lyon, Rennes, Strasbourg, and Toulouse.

In addition to that, there is also a lot of student monitoring and coaching on various projects. These activities vary from year to year so there are not detailed below, but they amount to tens of hours of additional pedagogical activities every year.

In Brief

- Academic research initiation module national coordinator
- National lecture coordinator
- 600+ undergraduate students
- 116h lectures + 42h lab. sess. = 158h/year

⇒ Full time equivalent = 200h/year

Adjunct lecturer at Sorbonne University/Upmc, Enst, Ensta, the French National "Beaux Arts" school…

[1] https://www.didierverna.net/lectures/

Title		Lectures	Lab. sessions	Since
Igpp	General Introduction to Programming Paradigms	2h		2016
Aop 1	Classical Object-Oriented Approach to Programming	12h	3 (2h) = 6h	2016
Afp	Functional Approaches to Programming	12h	4 (2h) = 8h	2008
Aop 2	Advanced Object-Oriented Approach to Programming	12h		2017
Typo	Typography and Numerical Typesetting	12h		2023

The workload is expressed as hours per student

2.1.1 Syllabus

Igpp – General Introduction to Programming Paradigms This is a general introduction to the concept of "programming paradigm", serving as a preamble to the following three subsequent courses: Aop 1, Aop 2, and Afp. We discuss the birth of computer science, the emergence of the concept of "paradigm" in the historical context of imperative and procedural programming, and the Von Neumann architecture. We also address the linguistic aspects of that notion (expressivity, tight link between language and thought, *etc.*).

Aop 1 – Classical Object-Oriented Approach to Programming This course is about the classical (industrial) object-oriented approach. This is notably the approach of the Java and C++ languages, which are both used for illustration purposes. The characteristics of this approach are the concepts of "class" for static information structuring, "message passing" (single dispatch) for dynamic behavior modelling, all of this in the general context of statically typed languages.
We detail the historical context in which the paradigm was born, its evolution and development, as well as the fundamental concepts on which it is built. We tackle the general modelling principles of this approach, and we also exhibit its limitations, putting emphasis on the tight relationship with the underlying programming languages characteristics.

00. Introduction Reminder on imperative and procedural programming. Genesis, history, evolution and development of the object-oriented approach. Current state of affairs.

01. Classes and Objects Class-based modelling. Concept of "object", instantiation, destruction (*vs.* garbage-collected languages). uml diagrams. Information scope (instance or class-wide attributes and methods) and accessibility (public or private attributes and methods).

02. Static Structuring Relations between classes: aggregation, composition, inheritance. Inheritance specificities: sub-classing *vs.* sub-typing, class hierarchies (simple and multiple inheritance), abstract and final classes, information accessibility (protected attributes and methods). Problems related to inheritance: inheritance *vs.* instantiation, inheritance *vs.* aggregation, ambivalence, diamond inheritance, public and private inheritance.

03. Dynamic Behavior Static polymorphism: overloading, masking. Dynamic polymorphism: overriding (virtual methods). Abstract methods. Interfaces *vs.* multiple inheritance. Limitations of the pseudo-equivalence sub-class $\Leftrightarrow$ sub-type: covariance, contravariance, Liskov substitution principle.

Afp – Functional Approaches to Programming The functional programming paradigm is very ancient, grounded in solid theoretical foundations, yet much less popular than the object-oriented one, although it has gained momentum in the past two decades. This course presents the functional programming paradigm in the broader sense, that is, not limited to the purely functional version. For illustration purposes, we use two languages in parallel: Lisp (the father of the paradigm) and Haskell (a more recent one). Both of these languages are functional, yet completely different in practically every aspect. This is precisely what makes their juxtaposition interesting.
We study the historical context in which the paradigm was born, the applicative principles it offers (primarily higher order functions), and the different language contexts in which this paradigm can be integrated: static vs. dynamic typing, pure or impure (with or without side effects), and with applicative or normal order evaluation.

00. Introduction History, evolution, and development of the paradigm: Lisp and the lambda-calculus, other languages. Comparison with imperative programming. Characteristics of the functional approach: higher order, pure or impure functional programming (with or without side effects), evaluation schemes (normal or applicative order).

01. Lisp and Haskell Differences tutorial: syntactic elements, concept of "expression" and "evaluation", interactivity. Dynamic and static typing. Conditionals, arithmetics, lists. Homoiconicity, reflexivity, meta-programming. Final comparison: 4 traits of the functional approach on the Newton-Raphson example.

02. Higher Order Reminders and generalities: definition, naming, assignment, locality. Anonymous functions. Functional arguments and design patterns: mapping, filtering, folding (reduction). Functional return values and design patterns: composition, operator cut, partial application. Functional objects: functional data structures and how they relate to the object-oriented paradigm.

04. Evaluation and Scoping Evaluation schemes: normal and applicative orders, consequences on the languages semantics. Lexical and dynamic scoping, block structure, locality, lexical closures, hygiene, name capture and name clash.

AOP 2 – Advanced Object-Oriented Approach to Programming This course is about an advanced, alternative approach to object orientation as described in AOP 1. Although still based on the concept of "class", this approach features a number of distinctive traits, in particular: dynamic typing, multiple dispatch, and reflexivity through a meta-object protocol. We aim to demonstrate that there is no single view on object orientation (not even on the class-based version), but also that the classical approach can be seen as a subset of this more general, hence, more expressive one.

01. CLOS – the Common Lisp Object System Introduction: history, distinctive traits. Classes and objects, instantiation, information scope and accessibility. Class – type integration. Inheritance: reminders and distinctive traits. Polymorphism: reminders, generic functions and methods. Covariance, contravariance, and multiple dispatch (multi-methods).

02. Advanced CLOS Meta-object protocols: method combinations, instantiation, `eql-specializers`. Meta-classes and applications: abstract, final, singleton, and counting classes. Reflexivity and application to the "circle-ellipse" problem.

03. Case Study: Binary Methods Comparison with the classical approach: covariance, contravariance. Resolution with the dynamic approach and multiple dispatch. User-level concept validation: introspection and meta-classes. Implementer-level concept validation: implementing an extension to the object layer.

04. Case Study: Design Patterns (the Visitor) Introduction and history of design patterns. The visitor example in C++. Naive reimplementation in Lisp. Deconstruction of the classical approach's idioms. Exhibition of traits that are functional rather than object-oriented. Extension to stateful visitors.

TYPO – Typography and Digital Typesetting This lecture is an introduction to the world of typography and digital typesetting, that is, the art of putting mostly textual documents into shape, and its history. Our goal is to provide a very broad view of the discipline, ranging from the scientific and technical, to the historical and esthetics angles. In particular, we emphasize on the relations between the beauty and the readability of text, the different typographical styles and usages, the tools allowing to compose documents of very high quality, and the underlying technical and algorithmic challenges.

01. History of Writing Proto-writing and writing systems: non-alphabetical and alphabetical, taxonomy (logograms, phonograms). Writing media: volumen and rotulus, codex, book, manuscript *vs.* printed, contemporary media (physical and digital).

02. History of Printing Prehistory: stamping, xylography, bloc printing. History: mobile characters in the far-east and in Europe. Press: flat-bed, cylindrical, rotary. Linotype, monotype, lumitype.

03. History of Fonts History: antique (medieval and humanistic influences, birth of roman), transition (modern roman), contemporary (slab-serif, sans-serif, "art nouveau"). Taxonomy: classifications, vox atypi. Anatomy of a character, ligatures, kerns.

04. The Digital Era Introduction: pre- and post-digital typographic worlds. First systems: dedicated hardware, timesharing and markup, contemporary typesetting systems. Tracing: segment-based, plotters, frames and bitmaps, by outline, ductus. Font formats: Adobe and Postscript, TEX and METAFONT, the font wars, Type 1, TrueType, OpenType, variable fonts.

05. Paragraph Justification: the Knuth-Plass Introduction and history. The TEX model: boxes, glues, esthetic criteria, tolerance, penalties, demerits. Problem modelling: solutions graph, shortest path, optimization with dynamic programming.

05. The Art of Composition Introduction: typographic criteria, objective, concept of "typographic color". Macro-typography: pagination rectangle, interlining. Ortho-typography: rules, conventions, customs, internationalization, multilingualization. Micro-typography: the TEX model, protrusion, expansion, letterspacing, wordspacing, washed-out, thieves, deceptive lines, orphans, widows, similarities.

2.2 Other Lectures

Title	Location	Type	Level	Workload	Period
The Music of Programs	Sorbonne University/UPMC	Conference	1st year Master's	2h	2025

2.3 Former Lectures

The following lectures are not given anymore. For the most part, the course materials are archived but continue to be publicly accessible. The number of students varied from around 50 (ENST, ENSTA, UPMC, French National "Beaux Arts" School) to 350 (EPITA).

Title	Location	Type	Level	Workload	Period
Image Synthesis	ENSTA/ ENST/ UPMC	Lectures + Lab. sessions	2nd year Master's	12h	2000 – 2010
Operating Systems	EPITA	Lectures	3rd year undergrad.	30h	2000 – 2017
Virtual Reality	ENST/ Beaux Arts	Lectures + Lab. sessions	3rd year undergrad.	9h	2000 – 2010
Parsing Techniques	EPITA	Lab. sessions	1st year undergrad.	15h	1999 – 2005
Lisp Programming	IONIS STM	Lab. sessions	MBA	15h	2011 – 2018

The hourly workload is given in hours per student

3 Visibility

In Brief

- Co-founder and president of the European Lisp Symposium (ELS) Steering Commitee
- DocEng (ACM SIGWEB Symposium on Document Engineering) Steering Committee member

---

- International conferences steering committees: 4 / 28 years cumulated
- International journal programme committees: 2 / 5 years cumulated
- International conferences chairs: 5 / 8 years cumulated
- International conferences programme committees: 11 / 32 years cumulated

3.1 International Conferences Steering Committees

- DocEng (ACM SIGWEB Symposium on Document Engineering), since 2026.
- Onward! (ACM SIGPLAN Symposium on New Ideas in Programming and Reflections on Software), 2020–2023.
- ELS (European Lisp Symposium), co-founder and president since 2008.
- ELW (European Lisp Workshop), 2007–2010.

3.2 International Journal Programme Committees

- Art, Science, and Engineering of Programming Journal, 2021.
- Journal of Universal Computer Science, 2007 – 2010.

3.3 International Conferences Chairs

- DocEng (ACM SIGWEB Symposium on Document Engineering), 2026.
- Onward! (ACM SIGPLAN Symposium on New Ideas in Programming and Reflections on Software) Essays, 2020.
- ILC (International Lisp Conference), 2014.
- ELS (European Lisp Symposium), 2011.
- ELW (European Lisp Workshop), 2007–2010.

3.4 Other Chairs

- Tutorial Chair, DocEng 2027 (ACM SIGWEB Symposium on Document Engineering)
- Local Chair, TUG 2027 (TEX Users Group Conference)
- Local Chair, ELS 2014 (European Lisp Symposium)

3.5 International Conferences Programme Committees

- DocEng (ACM SIGWEB Symposium on Document Engineering), 2025–....
- Onward! (ACM SIGPLAN Symposium on New Ideas in Programming and Reflections on Software) Essays, 2024.
- ICCQ (International Conference on Code Quality), 2023–...
- ILC (International Lisp Conference), 2012, 2014.
- DLS (Dynamic Languages Symposium), 2011, 2013, 2015.
- SAC (ACM SIGAPP Symposium on Applied Computing), 2012, 2013, 2015.
- ELS (European Lisp Symposium), 2011—2016.
- COP (Context-Oriented Programming Workshop), 2010, 2013, 2016, 2018.

- FARM (ACM SIGPLAN Functional Art, Music, Modelling, and Design Workshop), 2018.
- ELW (European Lisp Workshop), 2007—2010.
- DyLa (Dynamic Languages and Applications Workshop), 2013, 2014.

3.6 Other Committees

- Evaluation committee for the program Samuel De Champlain 2026–2027, section *Computer Science and Software and Electrical Engineering*, Quebec research fund for France-Quebec research teams mobility / FRQ – CFQCU.
- Most Notable Onward! 2010 Paper Award (selection committee 2020).
- Founder and first organizer of the Performance and Genericity Seminar of the Epita Research Lab (Lre) in 2008. Up to two sessions per month.

3.7 Book Technical Reviewer

- Lambda: the Ultimate Object – Object Orientation Elucidated. François-René Rideau. *2026.*
- The Act of Coding: Literary Aesthetics in Exploratory Programming, Christian Mehrstam. *CRC Press, Taylor & Francis Group, 2026.*
- GNU Autoconf, automake, libtool. Gary V. Vaughan, Ben Elliston, Tom Tromey and Ian Lance Taylor. *New Riders, October 2000.* Isbn 9781578701902.

4 Academic Tutoring Activities

In Brief

• Running Ph.D.'s	2
• Defended Ph.D.'s	2
• 2nd year Master's internships	3
• 1st year undergrad. internships	1
• Miscellaneous	1
• Student-Researchers	10
• Ph.D. jurys	4

4.1 Ph.D.'s

4.1.1 Running

- **Supervisor. Co-Tutoring 50%.**
  Maya Mouhammad. Network Intrusion Prediction by Federated and Interpretable Learning of Abnormalities in Distributed Environments. Epita/Lre, Sorbonne University/Upmc, APL Data Center, Edite of Paris.
- **Supervisor. Co-Tutoring 34%.**
  Khaoula Sghaier. Securing V2X Communication Exchanges in Telematic Control Units. Epita/Lre, Sorbonne University/Upmc, Telecom SudParis/SCN, VEDECOM, Edite of Paris.

4.1.2 Defended

- **Supervisor. Co-Tutoring 34%.**
  Baptiste Esteban. Implementing Modern and Efficient Morphological Techniques for Image Processing and Noise Estimation. Epita/Lre, Sorbonne University/Upmc, Edite of Paris. Defended on December 21st 2023.
- **Co-Tutoring 90%.**
  Jim E. Newton. Representing and Computing with Types in Dynamically Typed Languages. Epita/Lre, Sorbonne University/Upmc, Edite de Paris. Defended on November 20th 2018.

4.2 Master's Internships

- Clément Bonnefoy. Interface and Graphical Programming for an Image Processing Library. *STL Master's, Upmc, 2016.*
- Krista Druskhu. Comparative Language Study for the design of DSL's. *STL Master's, Upmc, 2015.*
- Vuong Ha Minh et Tung Nguyen Duc. Sustainability of a Benchmarking Platform for Common Lisp. *Software Engineering Master's, Bordeaux I University, 2010.*

4.3 Miscellaneous

- Julius Borghardt. Implementation and Application of Method Combinations for ERP-System Development. *Bachelor Thesis, TUHH/STS (Hamburg University of Technology, Software Technology Systems), 2025.*

4.4 Student-Researchers

The LRE includes undergraduate students for the duration of their scholarship. We train them for research by making them participate in our research projects. This is done in parallel with the first part of the regular scholarship and leads to a research-oriented major for the last part. They get several small research projects to conduct and validate under our supervision at LRE. Therefore, they can truly be considered "mini Ph.D. students". For proof, the stars (⋆) in the list below indicate those who have published before their graduation from EPITA (their names can be found in the "Publications" section). Finally, those who have pursued their studies up to a Master's and/or a Ph.D. are also indicated.

- Julie Viossat
- Aline Vongsavanh
- ⋆ Antoine Hacquard
- ⋆ Léo Valais
- ⋆ François Ripault
- ⋆ Laurent Senta
- ⋆ Christopher Chedeau
- Simon Odou, Master's and Ph.D. (LRI / Paris-Sud XI)
- ⋆ Guillaume Pitel, Master's and Ph.D. (LIMSI, Orsay)
- ⋆ Yoann Fabre, Master's and Ph.D. (UPMC)

4.5 Ph.D. Jurys and Supervisory Committees

- **Ph.D. Jury / reviewer.** Marco Heisig. Design and Implementation of the Parallel Programming Language Petalisp. *FAU Erlangen, Germany. July 2025. Supervisor: Harald Köstler.*
- **Ph.D. Jury / reviewer.** Jorge Vallejos. Modularising Context Dependency and Group Behaviour in Ambient-Oriented Programming Languages. *Software Languages Lab, Vrije Universiteit Brussel, July 2011. Supervision: Theo D'Hondt, Wolfgang De Meuter, Pascal Costanza.*
- **Supervisory committee.** Amaury Curiel. Binary Decision Trees and Acyclical Directed Graphs. *Sorbonne University/UPMC/-LIP6/APR. Direction: Antoine Genitrini.*
- **Supervisory committee.** Bertrand Petit-Heidelein. Diffuse Computing and Live Arts: Massively Interactive Music. *INRIA/CIRM, STIC (doctoral school). Supervision: Manuel Serrano, François Paris.*

5 Publications

In Brief

• Keynotes and invitations	6
• Book chapters	1
• Technical book reviewing	3
• International journals	4
• International conferences	39
• International workshops	4
• National conferences	6
• Others	9
• Talks	14

5.1 Book Chapters

1. Extensible languages: Blurring the distinction between DSLs and GPLs. Didier Verna. In Marjan Mernik, editor, *Formal and Practical Aspects of Domain-Specific Languages: Recent Developments*, chapter 1. IGI Global, September 2012. ISBN 9781466620926. DOI 10.4018/978-1-4666-2092-6.ch001.

5.2 Journal Articles

2. A theoretical and numerical analysis of the worst-case size of reduced ordered binary decision diagrams. Jim E. Newton and Didier Verna. *ACM Transactions on Computational Logic*, 20(1), January 2019. ISSN 15293785. DOI 10.1145/3274279.
3. Lisp, Jazz, Aikido. Didier Verna. *The Art, Science and Engineering of Programming Journal*, 2(3), March 2018. DOI 10.22152/programming-journal.org/2018/2/10.
4. Revisiting the visitor: the just do it pattern. Didier Verna. *Journal of Universal Computer Science*, 16(2):246–271, 2010. DOI 10.3217/jucs-016-02-0246.
5. Binary methods programming: the CLOS perspective. Didier Verna. *Journal of Universal Computer Science*, 14(20): 3389–3411, 2008. DOI 10.3217/jucs-014-20-3389.

5.3 International Conferences Articles

6. Quantum boundaries for last line adjustment. Didier Verna. In Barbara Beeton and Karl Berry, editors, *TUGboat*, volume 47, pages 267–276. TEX Users Group, TEX Users Group, September 2026. DOI 10.47397/tb/47-2/tb146verna-lastline.
7. An update on the method combinations MOP. Didier Verna. In *ELS 2026, the 19th European Lisp Symposium*, Krakow, Poland, May 2026. DOI 10.5281/zenodo.20024295.
8. Otarq: Adaptive and automated risk quantification method for ota updates in sdvs. Khaoula Sghaier, Ghada Gharbi, Badis Hammi, Pierre Merdrignac, Pierre Parrend, and Didier Verna. In *International Conference on Privacy, Security, and Trust (PST)*. IEEE, August 2026.

9. Dual-path v2n/v2v ota update distribution for sdvs: Design and experimental evaluation. Khaoula Sghaier, Arslane Hamza-Cherif, Pierre Merdrignac, Ghada Gharbi, Chahrazed Ksouri, Bechir Yengui, Badis Hammi, Pierre Parrend, and Didier Verna. In *Proc. IEEE Vehicular Technology Conference (VTC)*. IEEE, April 2026. DOI FIXME.

10. Towards more homogeneous paragraphs. Didier Verna. In *Proceedings of the ACM Symposium on Document Engineering 2025*, DocEng'25, pages 112–121, New York, NY, USA, September 2025. Association for Computer Machinery. ISBN 9798400711695. DOI 10.1145/3704268.3742696.

11. Similarity problems in paragraph justification: an extension to the Knuth-Plass algorithm. Didier Verna. In *Proceedings of the ACM SIGWEB Symposium on Document Engineering 2024*, DocEng'24, pages 127–130, New York, NY, USA, August 2024. Association for Computing Machinery. ISBN 9798400711695. DOI 10.1145/3685650.3685666.

12. A large scale format compliance checker for TEX font metrics. Didier Verna. In Barbara Beeton and Karl Berry, editors, *TUG'24, 42nd TEX Users Group Conference*, volume 45, pages 221–226. TEX Users Group, TEX Users Group, September 2024. DOI 10.47397/tb/45-2/tb140verna-tfm.

13. The Quickref cohort. Didier Verna. In *ELS'24, 17th European Lisp Symposium*, Vienna, Austria, May 2024. ISBN 9782955747483. DOI 10.5281/zenodo.10947962.

14. Structural analysis of the additive noise impact on the α-tree. Baptiste Esteban, Guillaume Tochon, Edwin Carlinet, and Didier Verna. In *Proceedings of the 20th International Conference on Computer Analysis of Images and Patterns (CAIP)*, volume 14185 of *Lecture Notes in Computer Science Series*, Limassol, Cyprus, September 2023. Springer.

15. Interactive and real-time typesetting for demonstration and experimentation: ETAP. Didier Verna. In Barbara Beeton and Karl Berry, editors, *TUG'23, 41st TEX Users Group Conference*, volume 44, pages 242–248. TEX Users Group, TEX Users Group, 2023. DOI 10.47397/tb/44-2/tb137verna-realtime.

16. A MOP-based implementation for method combinations. Didier Verna. In *ELS'23, 16th European Lisp Symposium*, pages 6–15, Amsterdam, Netherlands, April 2023. ISBN 9782955747476. DOI 10.5281/zenodo.7818680.

17. The cost of dynamism in static languages for image processing. Baptiste Esteban, Edwin Carlinet, Guillaume Tochon, and Didier Verna. In *Proceedings of the 21st ACM SIGPLAN International Conference on Generative Programming: Concepts & Experiences*, Auckland, New Zealand, December 2022.

18. Estimation of the noise level function for color images using mathematical morphology and non-parametric statistics. Baptiste Esteban, Guillaume Tochon, Edwin Carlinet, and Didier Verna. In *Proceedings of the 26th International Conference on Pattern Recognition*, Montréal, Québec, August 2022.

19. ETAP: Experimental typesetting algorithms platform. Didier Verna. In *ELS'22, 15th European Lisp Symposium*, pages 48–52, Porto, Portugal, March 2022. ISBN 9782955747469. DOI 10.5281/zenodo.6334248.

20. A corpus processing and analysis pipeline for quickref. Antoine Hacquard and Didier Verna. In *14th European Lisp Symposium*, pages 27–35, Online, May 2021. ISBN 9782955747452. DOI 10.5281/zenodo.4714443.

21. Finite automata theory based optimization of conditional variable binding. Jim E. Newton and Didier Verna. In *ELS'19, 12th European Lisp Symposium*, pages 26–33, Genova, Italy, April 2019. ISBN 9782955747438. DOI 10.5281/zenodo.2635402.

22. Implementing baker's `subtypep` decision procedure. Léo Valais, Jim E. Newton, and Didier Verna. In *ELS'19, 12th European Lisp Symposium*, pages 12–19, Genova, Italy, April 2019. ISBN 9782955747438. DOI 10.5281/zenodo.2646982.

23. Parallelizing Quickref. Didier Verna. In *ELS'19, 12th European Lisp Symposium*, pages 89–96, Genova, Italy, April 2019. ISBN 9782955747438. DOI 10.5281/zenodo.2632534.

24. Quickref: Common Lisp reference documentation as a stress test for Texinfo. Didier Verna. In Barbara Beeton and Karl Berry, editors, *TUG'19, 40th TEX Users Group Conference*, volume 40, pages 119–125. TEX Users Group, TEX Users Group, September 2019.

25. Strategies for typecase optimization. Jim E. Newton and Didier Verna. In *ELS'18, 11th European Lisp Symposium*, pages 23–31, Marbella, Spain, April 2018. ISBN 9782955747421. DOI 10.5281/zenodo.3405191.

26. Method combinators. Didier Verna. In *ELS'18, 11th European Lisp Symposium*, pages 32–41, Marbella, Spain, April 2018. ISBN 9782955747421. DOI 10.5281/zenodo.3247610.

27. Programmatic manipulation of Common Lisp type specifiers. Jim E. Newton, Didier Verna, and Maximilien Colange. In *ELS'17, 10th European Lisp Symposium*, pages 28–35, Vrije Universiteit Brussel, Belgium, April 2017. ISBN 9782955747414. DOI 10.5281/zenodo.3405363.

28. Type-checking of heterogeneous sequences in Common Lisp. Jim E. Newton, Akim Demaille, and Didier Verna. In *ELS'16, 9th European Lisp Symposium*, pages 13–20, AGH University of Science and Technology, Krakow, Poland, April 2016. ISBN 9782955747407. DOI 10.5281/zenodo.3405173.

29. The incredible tale of the author who didn't want to do the publisher's job. Didier Verna. In Barbara Beeton and Karl Berry, editors, *TUG'13, 34th TEX Users Group Conference*, volume 34. TEX Users Group, 2013.

30. TiCL: the prototype (Star TEX: the next generation, season 2). Didier Verna. In Barbara Beeton and Karl Berry, editors, *TUG'13, 34th TEX Users Group Conference*, volume 34. TEX Users Group, 2013.

31. Generic image processing with Climb. Laurent Senta, Christopher Chedeau, and Didier Verna. In *ELS'12, 5th European Lisp Symposium*, Zadar, Croatia, May 2012. DOI 10.5281/zenodo.3248934.

32. Star TEX: the next generation. Didier Verna. In Barbara Beeton and Karl Berry, editors, *TUG'12, 33rd TEX Users Group Conference*, volume 33. TEX Users Group, 2012.

33. Biological realms in computer science. Didier Verna. In *Onward!'11: the ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software Proceedings*, pages 167–176. ACM, October 2011. ISBN 9781450309417. DOI 10.1145/2089131.2089140.

34. Towards LATEX coding standards. Didier Verna. In Barbara Beeton and Karl Berry, editors, *TUG'11, 32nd TEX Users Group Conference*, volume 32, pages 309–328. TEX Users Group, 2011.

35. CLoX: Common lisp objects for XEmacs. Didier Verna. In *ELS'10, 3rd European Lisp Symposium*, Lisbon, Portugal, May 2010. DOI 10.5281/zenodo.3248958.

36. Classes, styles, conflicts: the biological realm of LATEX. Didier Verna. In Barbara Beeton and Karl Berry, editors, *TUG'10, 31st TEX Users Group Conference*, volume 31, pages 162–172. TEX Users Group, 2010.

37. CLOS efficiency: Instantiation. Didier Verna. In *ILC'09 International Lisp Conference*, pages 76–90, MIT, Cambridge, Massachusetts, USA, March 2009. ALU (Association of Lisp Users). DOI 10.5281/zenodo.3386206.

38. Binary methods programming: the CLOS perspective. Didier Verna. In *ELS'08, 1st European Lisp Symposium*, pages 91–105, Bordeaux, France, May 2008. DOI 10.5281/zenodo.3248977.

39. Action recognition: How intelligent virtual environments can ease human-machine interaction. Didier Verna. In Hal Thwaites and Scott Thrane Refsland, editors, *VSMM'00, Sixth International Conference on Virtual Systems and MultiMedia*, pages 703–713, Gifu Research and Development Foundation, Gifu, Japan, October 2000. International Society on Virtual Systems and MultiMedia, Ohmsha Press.

40. *Urbi et Orbi*: Unusual design and implementation choices for distributed virtual realities. Didier Verna, Yoann Fabre, and Guillaume Pitel. In Hal Thwaites and Scott Thrane Refsland, editors, *VSMM'00, Sixth International Conference on Virtual Systems and MultiMedia*, pages 714–724, Gifu Research and Development Foundation, Gifu, Japan, October 2000. International Society on Virtual Systems and MultiMedia, Ohmsha Press.

41. The multicast support in XEmacs. Didier Verna. In *m17n'99, 3rd International Symposium on Multilingual Environments*, Tsukuba, Japan, 1999.

42. Ergonomics and human-machine interaction concerns in Mule. Didier Verna. In *m17n'99, 3nd International Symposium on Multilingual Environements*, Tsukuba, Japan, 1999.

43. Can we define virtual reality? the MrIC model. Didier Verna and Alain Grumbach. In Jean-Claude Heudin, editor, *Virtual Worlds 98*, Lecture Notes in Artificial Intelligence, pages 29–41. Springer-Verlag, 1998.

44. Télé-opération et réalité virtuelle: Assistance à l'opérateur par modélisation cognitive de ses intentions. Didier Verna. In *IHM'97*, pages 205–206. Cépaduès-Éditions, 1997.

5.4 International Workshops Articles

45. Recognizing heterogeneous sequences by rational type expression. Jim E. Newton and Didier Verna. In *Meta'18, Meta-Programming Techniques and Reflection Workshop*, Boston, MA, USA, November 2018.

46. Context-oriented image processing. Didier Verna and François Ripault. In *COP'15, Context-Oriented Programming Workshop*, 2015. ISBN 9781450336543. DOI 10.1145/2786545.2786547.

47. Beating C in scientific computing applications. Didier Verna. In *ELW'06, 3rd European Lisp Workshop*, Nantes, France, July 2006.

48. Augmented reality, the other way around. Didier Verna and Alain Grumbach. In M. Gervautz, A. Hildebrand, and D. Schmalstieg, editors, *EGVE, 5th Eurographics Workshop on Virtual Environments*, pages 147–156. Springer, 1999.

5.5 National Conferences Articles

49. Analyse des performances d'arbres de décision pour la détection d'intrusions cyber et d'anomalies dans les réseaux informatiques. Maya Mouhamad, Ali Alaeldine, Georges. Ouffoue, Nidà Meddouri, Didier Verna, and Pierre Parrend. In *Proceeding of the first Symposium Doctoral Francophone en Intelligence Artificielle*, October 2025.

50. Approche automatisée pour l'évaluation des risques des mises à jour ota dans les véhicules définis par logiciel. Khaoula Sghaier, Ghada Gharbi, Badis Hammi, Pierre Merdrignac, Pierre Parrend, and Didier Verna. In *Conférence RESSI*, April 2026.

51. Analyse structurelle de l'influence du bruit sur l'arbre alpha. Baptiste Esteban, Guillaume Tochon, Edwin Carlinet, and Didier Verna. In *29e Colloque sur le traitement du signal et des images*, Grenoble, France, August 2023. GRETSI - Groupe de Recherche en Traitement du Signal et des Images.

52. Estimation de la fonction de niveau de bruit pour des images couleurs en utilisant la morphologie mathématique. Baptiste Esteban, Guillaume Tochon, Edwin Carlinet, and Didier Verna. In *Proceedings of the 28st Symposium on Signal and Image Processing (GRETSI)*, Nancy, France, September 2022.

53. Généricité dynamique pour des algorithmes morphologiques. Baptiste Esteban, Edwin Carlinet, Guillaume Tochon, and Didier Verna. In *Proceedings of the 28st Symposium on Signal and Image Processing (GRETSI)*, Nancy, France, September 2022.

54. Définir le virtuel: une vision cognitive. Didier Verna. In *ReViCo'99, Réalité Virtuelle et Cognition*, Paris, France, December 1999.

5.6 Keynotes and Invitations

55. The art, science, and craftsmanship of typography. Didier Verna. Keynote at <Programming> 2026, March 2026.

56. Traitement des similarités dans la justification de paragraphe. Didier Verna. Exposé GUTenberg, September 2025.

57. Justification de paragraphe: le Knuth-Plass. Didier Verna. Exposé GUTenberg, January 2024.

58. Biological realms in computer science. Didier Verna. Keynote at ACCU'14, apr 2014.

59. CLOS efficiency: Instantiation. Didier Verna. Invited Talk at the Vrije University of Brussels, 2010.

60. Scientific computing in lisp: Beyond the performance of C. Didier Verna. Invited Talk at LaBRI, Université de Bordeaux I, France, 2006.

5.7 Reports and Various Publications

61. *(Dynamic (Programming Paradigms)) ;; Performance and Expressivity.* Didier Verna. PhD thesis, EDITE de Paris, Sorbone-Université, EPITA/LRDE, July 2020. Habilitation à Diriger les Recherches.

62. JSPP: Morphing C++ into JavaScript. Christopher Chedeau and Didier Verna. Technical Report 201201-TR, LRDE (EPITA Research and Development Laboratory), January 2012.

63. Standard output streams default behavior in terminal sessions. Didier Verna. Common Document Repository #11, 2012. Doi 10.5281/zenodo.3414042.

64. Clarification proposal for CLHS 22.3. Didier Verna. Common Document Repository #7, 2011. Doi 10.5281/zenodo.3413913.

65. File-local variables. Didier Verna. Common Document Repository #9, 2011. Doi 10.5281/zenodo.3414042.

66. LaTeX curricula vitae with the CurVe class. Didier Verna. *The PracTeX Journal*, (3), August 2006.

67. CV formatting with CurVe. Didier Verna. *TUGBoat, Communications of the TeX Users Group*, 22(4):361–364, December 2001. Issn 0896320.

68. *Télé-opération et Réalité Virtuelle: Assistance à l'Opérateur par Modélisation Cognitive de ses Intentions.* Didier Verna. PhD thesis, ENST (École Nationale Supérieure des Télécommunications de Paris), Paris, France, February 2000. ENST 00 E007.

69. Comment définir le virtuel ? le modèle MrIC. Didier Verna. Technical Report 97 D 008, ENST (École Nationale Supérieure des Télécommunications de Paris), 46 rue Barrault, 75013 Paris, France, 1997.

5.8 Talks

70. A taste of Julia. Didier Verna. ACCU'16, April 2016.
71. A taste of Julia. Didier Verna. Séminaire Performance et Généricité du LRDE, April 2016.
72. La musique des programmes. Didier Verna. Soirée Thématique de l'EPITA: l'Esthétique en Informatique, 2016.
73. Referential transparency is overrated. Didier Verna. ACCU'15, April 2015.
74. The bright side of exceptions. Didier Verna. ACCU'13, April 2013.
75. Extensibility for DSL design and implementation: a case study in Lisp. Didier Verna. DSLDI'13, DSL Design and Implementation Workshop, April 2013.
76. DSLs from the perspective of extensible languages. Didier Verna. ACCU'12, April 2012.
77. Lisp extensibility: Impact on DSL design and implementation. Didier Verna. Tutorial at ILC'12, the International Lisp Conference, 2012.
78. Meta-circularity, and vice-versa. Didier Verna. ACCU'11, April 2011.
79. Clon, the command-line options nuker. Didier Verna. ILC'10, the International Lisp Conference, 2010.
80. Revisiting the visitor: the just do it pattern. Didier Verna. ACCU'09, April 2009.
81. Performance and genericity: the forgotten power of Lisp. Didier Verna. ACCU'08, April 2008.
82. Sémantique et localisation de l'assistance en réalité virtuelle. Didier Verna and Alain Grumbach. In *Journées Nationales du Groupe de Travail sur la Réalité Virtuelle*, pages 105–112, 1998.
83. Assistance cognitive à la télé-opération en monde virtuel. Alain Grumbach and Didier Verna. In *Journées Nationales du Groupe de Travail sur la Réalité Virtuelle*, pages 38–46, 1996.