



Curriculum Vitæ

Didier Verna
EPITA/ LRE
14-16 rue Voltaire,
94276 Le Kremlin-Bicêtre

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



0000-0002-6315-052X



profile/Didier_Verna



in/didierverna

Distinctions

Reviewers Choice Award 2018 du Art, Science and Engineering of Programming Journal
(Verna, 2018a, n° 3)

Keynote Speaker à ACCU 2014
(Verna, 2014, n° 51)

Best Paper Award 2006 du European Lisp Workshop
(Verna, 2006a, n° 43)

Actuellement

Position : Enseignant-Chercheur habilité à diriger les recherches
Établissement d'enseignement : École pour l'Informatique et les Techniques Avancées (EPITA)
Établissement de Recherche : Laboratoire de Recherche de l'EPITA (LRE)

Parcours

2024 **Qualification Professeur des Universités**

Section: 27 — Informatique
Jury: Olga Kouchnarenko, Céline Rouveirol

2020 **Habilitation à Diriger les Recherches, EDITE de Paris, Sorbonne Université, EPITA/LRE**

Titre: (dynamic (programming paradigms)) ;; *performance and expressivity*
Jury: Manuel Serrano, INRIA, Sophia Antipolis, France (rapporteur)
Robert Strandh, Université de Bordeaux, France (rapporteur)
Nicolas Neuß, FAU, Erlangen-Nürnberg, Allemagne (rapporteur)
Marco Antoniotti, Université de Milan-Bicocca, Italie (examineur)
Ralf Möller, Université de Lübeck, Allemagne (examineur)
Gérard Assayag, IRCAM, Paris, France (examineur).

2000 **Docteur de l'École Nationale Supérieure des Télécommunications de Paris (ENST)**

Titre: Téléopération et Réalité Virtuelle — Assistance à l'Opérateur par Modélisation Cognitive de ses Intentions
Jury: Philippe Coiffet, Jean-François Richard (rapporteurs)
Jean-Paul Papin (examineur)
Alain Grumbach (directeur)

1994 Ingénieur de l'ENST, spécialisation Informatique et Réseaux
1988 Baccalauréat scientifique (C)

Expérience Professionnelle

2014–... Membre de l'EDITE de Paris (École Doctorale Informatique, Télécommunications et Électronique)
2005–2018 Enseignant vacataire à l'ENST, l'ENSTA, l'UPMC, l'École Nationale des Beaux Arts et IONIS STM
Cf. détails dans la partie « Activités d'Enseignement »
1999–... Enseignant-Chercheur à l'EPITA (CDI)

À Noter

Je travaille à temps partiel (3/5^e, puis récemment 4/5^e). Toutes les données chiffrées qui suivent sont exprimées en valeur absolue. Les heures d'enseignement sont accompagnées d'un calcul d'équivalent temps plein, mais le reste (densité de publication, taux de participation à l'écosystème, etc.) est à évaluer relativement à mon taux d'activité.

1 Responsabilités Administratives

2020–... Référent national cours d'informatique fondamentale, cf. section « Enseignement »
2009–... Participation au processus de recrutement des étudiants de l'EPITA (salons, JPO, oraux d'admission, etc.)
Environ 90h / an
2008–... Co-fondateur et président du comité de pilotage d'ELS (European Lisp Symposium)
2008 Fondateur et premier organisateur du séminaire Performance et Généricité du LRE

2 Activités d'Enseignement

Je suis le concepteur de tous les cours que je donne depuis 1999. Tous mes supports de cours sont disponibles publiquement sur la [page enseignement](#)¹ de mon site web scientifique, y compris les anciens cours archivés.

2.1 Actuellement à l'EPITA

Les cours ci-dessous concernent plus de 600 étudiants en tronc commun du cycle ingénieur (niveau L3) chaque année. J'en suis le « référent national » : je conçois et donne ces cours moi-même dans l'antenne parisienne de l'école (environ 400 étudiants / an en 3 groupes) et je gère par ailleurs des équipes de 4 à 6 enseignants qui les dispensent, avec mes supports, dans les quatre antennes régionales : Lyon, Rennes, Strasbourg, et Toulouse.

À cela se rajoutent de nombreux encadrements d'étudiants sur des projets divers et variés, du coaching, du suivi d'étudiants en filière apprentissage, et d'étudiants chercheurs (cf. Section 4.3 page 6). Ces activités étant très variables d'une année sur l'autre, elles ne sont pas détaillées ici, bien que représentant quelques dizaines d'heures de face à face pédagogique supplémentaires chaque année.

En Bref

- Référent national cours
 - 600+ étudiants/an niveau L3
 - 116h CM + 42h TP = 158h/an
- ⇒ Équivalent temps plein = 200h/an

Anciennement vacataire à l'ENST, l'ENSTA, l'UPMC, et l'École des Beaux Arts.

Intitulé		Cours Magistraux	TPs	Depuis
IGPP	Introduction Générale aux Paradigmes de Programmation	2h		2016
AOP 1	Approche Objet Classique de la Programmation	12h	3 (2h) = 6h	2016
AFP	Approches Fonctionnelles de la Programmation	12h	4 (2h) = 8h	2008
AOP 2	Approche Objet Avancée de la Programmation	12h		2017
TYPO	Typographie : Art, Artisanat, et Science	12h		2023

Le volume horaire est donné en heures par étudiant

2.1.1 Syllabus

IGPP – Introduction Générale aux Paradigmes de Programmation Il s'agit d'une introduction générale à la notion de paradigme de programmation, qui sert de préambule aux trois cours suivants : AOP 1, AOP 2, et AFP. On y évoque la naissance de l'informatique, l'émergence de la notion de paradigme dans le contexte historique de la programmation impérative et procédurale et de l'architecture de Von Neumann, ainsi que les aspects linguistiques et langagiers de cette notion (expressivité, lien entre langage et pensée, etc.).

1. <https://www.lrde.epita.fr/~didier/lectures/>

AOP 1 – Approche Objet Classique de la Programmation Ce cours traite de l’approche orientée objet classique (industrielle). C’est en particulier l’approche des langages Java et C++ qui sont tous deux utilisés en guise d’illustration. Les caractéristiques de cette approche sont la notion de « classe » pour la structuration statique de l’information, l’« envoi de message » pour la modélisation du comportement dynamique, le tout dans le contexte général des langages statiquement typés. On détaille le contexte historique dans lequel ce paradigme est né, son évolution et son développement, ainsi que les concepts fondamentaux au-dessus desquels il s’articule. On aborde les grands principes de modélisation dans cette approche, et on en montre également les limites, en mettant l’accent sur les liens étroits avec les caractéristiques des langages sous-jacents.

- 00. Introduction** Rappels sur les paradigmes impératifs et procéduraux. Genèse, histoire, évolution et développement de l’approche orientée objet. Situation actuelle.
- 01. Classes et Objets** Modélisation à base de classe. Notion d’objet, instanciation, destruction (vs. langages à ramasse-miettes). Diagrammes UML. Portée (attributs et méthodes d’instance ou de classe) et accessibilité (attributs et méthodes privées ou publiques) de l’information.
- 02. Structuration Statique** Relations entre classes : agrégation, composition, héritage. Caractéristiques de l’héritage : sous-classage vs. sous-typage, hiérarchies de classes (héritage simple et multiple), classes abstraites et finales, accessibilité de l’information (attributs et méthodes protégés). Problèmes liés à l’héritage : héritage vs. instanciation, héritage vs. agrégation, ambivalence, héritage en losange, héritage publique et privé.
- 03. Comportement Dynamique** Polymorphisme statique : surcharge, masquage. Polymorphisme dynamique : réécriture (méthodes virtuelles). Méthodes abstraites. Interfaces vs. héritage multiple. Limitations de la pseudo-équivalence sous-classe $\Leftrightarrow$ sous-type : covariance, contravariance, principe de substitution de Liskov.

AFP – Approches Fonctionnelles de la Programmation Le paradigme de programmation fonctionnelle est très ancien, solidement ancré dans des fondations théoriques, et pourtant beaucoup moins populaire que le paradigme objet, bien qu’il ait éprouvé un regain d’intérêt ces deux dernières décennies. Ce cours traite du paradigme de programmation fonctionnelle au sens large, c’est-à-dire non réduit au seul fonctionnel pur. En guise d’illustration, nous utilisons deux langages en parallèle : Lisp (le père du paradigme), et Haskell (un langage plus récent). Ces deux langages sont fonctionnels et pourtant complètement différents sur pratiquement tous leurs aspects. C’est précisément ce qui rend leur juxtaposition intéressante. On détaille le contexte historique dans lequel ce paradigme est né, les principes applicatifs qu’il offre (principalement les fonctions d’ordre supérieur) ainsi que leurs fondements théoriques. On met l’accent sur les différents contextes langagiers d’intégration de ce paradigme, et en particulier avec typage statique ou dynamique, avec ou sans effets de bords (pur / impur), et dans le cadre d’évaluateurs à l’ordre normal ou applicatif.

- 00. Introduction** Histoire, évolution et développement du paradigme : Lisp et le Lambda calcul, autres langages. Comparaison avec la programmation impérative. Caractéristiques de l’approche fonctionnelle : ordre supérieur, fonctionnel pur ou impur (avec ou sans effet de bord), formes d’évaluation (ordre normal ou applicatif).
- 01. Lisp et Haskell** Tutoriel des différences : éléments syntaxiques, notions d’expression et d’évaluation, interactivité. Typage dynamique et statique. Conditionnels, arithmétique, listes. Homoïconicité, réflexivité, méta-programmation. Comparatif final : 4 caractéristiques de l’approche fonctionnelle sur l’exemple de Newton-Raphson.
- 02. Ordre Supérieur** Rappels et généralités : définition, nommage, affectation, localité. Fonctions anonymes. Arguments fonctionnels et design patterns : mapping, filtrage, réduction. Retours fonctionnels et design patterns : composition, coupure d’opérateur, application partielle. Objets fonctionnels : structures de données fonctionnelles et lien avec le paradigme objet.
- 04. Évaluation et Scoping** Techniques d’évaluation : ordres normal et applicatif, conséquences sur la sémantique des langages. Scoping lexical et dynamique, structure de blocs, localité, fermetures lexicales, hygiène, capture, et collision.

AOP 2 – Approche Objet Avancée de la Programmation Ce cours traite d’une approche objet avancée, alternative à l’approche classique d’AOP 1. Bien que toujours basée sur la notion de classe, cette approche possède un certain nombre de traits distinctifs : typage dynamique, multi-méthodes, et réflexivité au travers d’un protocole méta-objet en sont les principaux. Le but est à la fois de montrer qu’il n’existe pas qu’une seule vision du paradigme objet, fût-il à base de classe, mais aussi que l’approche classique peut être vue comme un cas particulier de cette approche plus générale et donc plus expressive.

- 01. CLOS – le Common Lisp Object System** Introduction : historique, caractéristiques distinctives. Classes et objets, instanciation, portée et accessibilité de l’information. Intégration classe – type. Héritage : rappels et caractéristiques distinctives. Polymorphisme : rappels, fonctions génériques et méthodes. Covariance, contravariance et dispatch multiple (multi-méthodes).
- 02. CLOS Avancé** Protocoles méta-objet : combinaison de méthodes, instanciation, eql-specialiseurs. Méta-classes et application : classes abstraites, finales, singleton et comptables. Réflexivité et application : problème du « cercle elliptique ».
- 03. Étude de Cas : Méthodes Binaires** Comparaison avec l’approche classique : covariance, contravariance. Résolution avec l’approche dynamique et le dispatch multiple. Validation du concept niveau utilisateur : introspection et méta-classes. Validation du concept niveau implémenteur : méta-programmation d’une extension de la couche objet.

04. Étude de cas : Design Patterns (le Visiteur) Introduction et historique des design patterns. Exemple du visiteur en C++. Réimplémentation naïve en Lisp. Déconstruction des idiotismes de l'approche classique. Mise en évidence de motifs fonctionnels plutôt qu'orientés objet. Extension aux visiteurs à état.

Typo – Typographie : Art, Artisanat, et Science Ce cours est une introduction au monde de la typographie, c'est-à-dire l'art de mettre en forme des documents essentiellement textuels, et à son histoire. L'objectif est de fournir une vision très large de la discipline, à la fois sur les plans scientifiques, techniques, historiques, et esthétiques. On y aborde en particulier les liens entre la beauté et la lisibilité d'un texte, les styles et usages typographiques, les outils permettant de composer en très haute qualité, et les enjeux techniques et algorithmiques sous-jacents.

01. Histoire de l'Écriture Systèmes de proto-écriture et écriture : non alphabétiques et alphabétiques, taxonomie (logogrammes, phonogrammes). Supports d'écriture : volumen et rotulus, codex, livre, manuscrit vs. imprimé, supports contemporains (physiques et dématérialisés).

02. Histoire de l'Impression Préhistoire : estampillage, xylographie, impression par bloc. Histoire : caractères mobiles en extrême-orient et en Europe. Presses : lit-plat, cylindriques, rotatives. Linotypes, monotypes, lumitype.

03. Histoire des Fontes Histoire : antique (influences médiévales et humanistes, naissance du romain), transition (romain moderne), contemporain (égyptiennes, sans-serif, art nouveau). Taxonomie : classifications, vox atypi. Anatomie d'un caractère, chasse, ligature, crénage.

04. L'Ère Numérique Introduction : mondes typographiques pré- et post-numériques. Premiers systèmes : machines dédiés, temps partagé et markup, systèmes typographiques contemporains. Tracé de caractères : par segments, plotters, trames et bitmaps, par contour, ductus. Formats de polices : Adobe et PostScript, T_EX et METAFONT, guerre des polices, Type 1, TrueType, OpenType, polices variables.

05. Justification de Paragraphe : le Knuth-Plass Introduction et historique. Le modèle de T_EX : boîtes, glues, critères esthétiques, tolérance, pénalités, démerites. Modélisation du problème : graphe des solutions, plus court chemin, optimisation par programmation dynamique.

05. L'Art de la Composition Introduction : critères typographiques, finalité, notion de gris typographique. Macro-typographie : rectangle d'empagement, interlignage. Ortho-typographie : règles, conventions et usages, internationalisation, multilinguisation. Micro-typographie : modèle de T_EX, saillie, expansion, interlettrage, intermottage, lignes lavées, à voleur, trompeuses, veuves, orphelines, similarités.

2.2 Anciens Cours

Les cours suivants ne sont plus donnés. Pour la plupart, les supports de cours correspondants sont archivés mais continuent d'être accessibles publiquement. Le nombre d'étudiants concernés variait d'une cinquantaine (ENST, ENSTA, UPMC, Beaux Arts) à 350 (EPITA) en plusieurs groupes.

Intitulé	Lieu	Type	Niveau	Volume	Période
Synthèse d'Images	ENSTA/ ENST/ UPMC	CM + TD	M2	12h	2000 – 2010
Systèmes d'Exploitation	EPITA	CM	L3	30h	2000 – 2017
Réalité Virtuelle	ENST/ Beaux Arts	CM + TD	L3	9h	2000 – 2010
Analyse Lexico-Syntaxique	EPITA	TD	L1	15h	1999 – 2005
Programmation Lisp	IONIS STM	TD	MBA	15h	2011 – 2018

Le volume horaire est donné en heures par étudiant

3 Visibilité

3.1 Comités de Pilotage de Conférences Internationales

- Onward! (ACM SIGPLAN Symposium on New Ideas in Programming and Reflections on Software), 2020–2023.
- ELS (European Lisp Symposium), co-fondateur et président depuis 2008.
- ELW (European Lisp Workshop), 2007–2010.

3.2 Comités de Lecture de Revues Internationales

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

3.3 Chaires de Conférences Internationales

- 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 Comités de programme de conférences internationales

- 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), 2023–2025.
- 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.

3.5 Autres comités

- Most Notable Onward! 2010 Paper Award (comité de sélection 2020).
- Fondateur et premier organisateur du séminaire Performance et Généricité du LRE en 2008. Une à deux sessions par mois.

3.6 Recteur Technique de Livre

- GNU Autoconf, automake, libtool. Gary V. Vaughan, Ben Elliston, Tom Tromey and Ian Lance Taylor. *New Riders, October 2000*. ISBN 9781578701902.

En Bref

Co-fondateur et président du comité de pilotage d'ELS (European Lisp Symposium)

- Comités de pilotage de conférences internationales: 3 / cumul 22 ans
- Comités de lecture de revues internationales: 2 / cumul 5 ans
- Chaires de conférences internationales: 5 / cumul 8 ans
- Comités de programme de conférences internationales: 11 / cumul 30 ans

4 Activités d'Encadrement

4.1 Thèses

4.1.1 En Cours

- **Directeur. Co-encadrement 50%.**
Maya Mouhammad. Prédiction des Intrusions Réseau par Apprentissage Fédéré et Interprétable des Anomalies dans un Environnement Distribué. EPITA/LRE, Sorbonne Université/UPMC, APL Data Center, EDITE de Paris.
- **Directeur. Co-encadrement 34%.**
Khaoula Sghaier. Sécurisation des Échanges de Communication V2X des Unités de Contrôle Télématiques. EPITA/LRE, Sorbonne Université/UPMC, Telecom SudParis/SCN, VEDECOM, EDITE de Paris.

4.1.2 Soutenues

- **Directeur. Co-encadrement 34%.**
Baptiste Esteban. Implémentations Modernes et Performantes de Techniques Morphologiques pour le Traitement d'Images et l'Estimation de Bruit. EPITA/LRE, Sorbonne Université/UPMC, EDITE de Paris. Soutenue le 21 Décembre 2023.
- **Co-encadrement 90%.**
Jim E. Newton. Representing and Computing with Types in Dynamically Typed Languages. EPITA/LRE, Sorbonne Université/UPMC, EDITE de Paris. Soutenue le 20 Novembre 2018.

4.2 Stages de Master

- Clément Bonnefoy. Interface et programmation graphique pour une bibliothèque de traitement d'images. *Master STL, UPMC, 2016.*
- Krista Drusku. Étude comparative de langages pour la conception de DSL. *Master STL, UPMC, 2015.*
- Vuong Ha Minh et Tung Nguyen Duc. Pérénsation d'une plateforme de benchmarking pour Common Lisp. *Master Génie Logiciel, Université de Bordeaux I, 2010.*

4.3 Étudiants-Chercheurs

Le LRE intègre des étudiants pour la durée de leur cursus ingénieur. Nous leur dispensons une formation par la recherche en les faisant participer à nos projets. Cela s'effectue en parallèle du cursus classique pour le tronc commun et cela devient formalisé par une majeure « double compétence orientée recherche » en fin d'études. Ils ont donc un ou plusieurs mini-sujets de recherche à mener et à valider sous notre supervision durant leur passage au LRE. Ces étudiants peuvent véritablement être considérés comme de « mini-thésards ». Pour s'en convaincre, dans la liste ci-dessous, les étoiles (★) indiquent ceux qui ont été publiant avant leur sortie d'EPITA (donc, avant même leur 3^e cycle). On retrouve ainsi leurs noms dans la section Publications. Enfin, le cas échéant, ceux dont le parcours scolaire a continué sur un DEA (pour les plus anciens), un Master (plus les plus jeunes) puis une thèse, sont également indiqués.

- Aline Vongsavanh
- ★ Antoine Hacquard
- ★ Léo Valais
- ★ François Ripault
- ★ Laurent Senta
- ★ Christopher Chedeau
- Simon Odou, Master et thèse (LRI / Paris-Sud XI)
- ★ Guillaume Pitel, DEA SC et thèse (LIMSI, Orsay)
- ★ Yoann Fabre, DEA IARFA et thèse (UPMC)

4.4 Jurys de thèse et comités de suivi

- **Jury de thèse / rapporteur.** Marco Heisig. Design and Implementation of the Parallel Programming Language Petalisp. *FAU Erlangen, Allemagne. Juillet 2025. Directeur: Harald Köstler.*
- **Jury de thèse / rapporteur.** Jorge Vallejos. Modularising Context Dependency and Group Behaviour in Ambient-Oriented Programming Languages. *Software Languages Lab, Vrije Universiteit Brussel, Juillet 2011. Direction: Theo D'Hondt, Wolfgang De Meuter, Pascal Costanza.*
- **Comité de suivi.** Amaury Curiel. Arbres de Décision Binaires et Graphes Dirigés Acycliques. *Sorbonne Université/UPMC/LIP6/APR.*
- **Comité de suivi.** Bertrand Petit-Heidelein. Informatique Diffuse et Arts de la Scène: Musique Massivement Interactive. *INRIA/CIRM, STIC (école doctorale). Direction: Manuel Serrano, François Paris.*

En Bref

- Thèses en cours 2
- Thèses soutenues 2
- Stages de M2 3
- Stages de M1 1
- Étudiants-Chercheurs 9
- Jurys de thèses / suivi 3

• Keynotes et invitations	5
• Chapitres de livre	1
• Relecture technique de livre	1
• Revues internationales	4
• Conférences internationales	35
• Workshops internationaux	4
• Conférences francophones	4
• Autres	9
• Exposés Oraux	14

5 Publications

5.1 Chapitres de Livres

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](https://doi.org/10.4018/978-1-4666-2092-6.ch001).

5.2 Articles de Revues

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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/10.3217/jucs-014-20-3389).

5.3 Articles de Conférences Internationales

6. 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](https://doi.org/10.1145/3704268.3742696).
7. 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](https://doi.org/10.1145/3685650.3685666).
8. A large scale format compliance checker for $\text{\TeX}$ font metrics. Didier Verna. In Barbara Beeton and Karl Berry, editors, *TUG’24, 42nd $\text{\TeX}$ Users Group Conference*, volume 45, pages 221–226. $\text{\TeX}$ Users Group, $\text{\TeX}$ Users Group, September 2024. DOI [10.47397/tb/45-2/tb140verna-tfm](https://doi.org/10.47397/tb/45-2/tb140verna-tfm).
9. The Quickref cohort. Didier Verna. In *ELS’24, 17th European Lisp Symposium*, Vienna, Austria, May 2024. ISBN 9782955747483. DOI [10.5281/zenodo.10947962](https://doi.org/10.5281/zenodo.10947962).
10. 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.
11. Interactive and real-time typesetting for demonstration and experimentation: ETAP. Didier Verna. In Barbara Beeton and Karl Berry, editors, *TUG’23, 41st $\text{\TeX}$ Users Group Conference*, volume 44, pages 242–248. $\text{\TeX}$ Users Group, $\text{\TeX}$ Users Group, 2023. DOI [10.47397/tb/44-2/tb137verna-realtime](https://doi.org/10.47397/tb/44-2/tb137verna-realtime).
12. 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](https://doi.org/10.5281/zenodo.7818680).
13. 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.
14. 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.
15. 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](https://doi.org/10.5281/zenodo.6334248).
16. 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](https://doi.org/10.5281/zenodo.4714443).
17. 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](https://doi.org/10.5281/zenodo.2635402).
18. 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](https://doi.org/10.5281/zenodo.2646982).
19. 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](https://doi.org/10.5281/zenodo.2632534).

20. Quickref: Common Lisp reference documentation as a stress test for Texinfo. Didier Verna. In Barbara Beeton and Karl Berry, editors, *TUG'19, 40th T_EX Users Group Conference*, volume 40, pages 119–125. T_EX Users Group, T_EX Users Group, September 2019.
21. 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](https://doi.org/10.5281/zenodo.3405191).
22. 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](https://doi.org/10.5281/zenodo.3247610).
23. 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](https://doi.org/10.5281/zenodo.3405363).
24. 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](https://doi.org/10.5281/zenodo.3405173).
25. 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 T_EX Users Group Conference*, volume 34. T_EX Users Group, 2013.
26. TiCL: the prototype (Star T_EX: the next generation, season 2). Didier Verna. In Barbara Beeton and Karl Berry, editors, *TUG'13, 34th T_EX Users Group Conference*, volume 34. T_EX Users Group, 2013.
27. Generic image processing with Climbl. Laurent Senta, Christopher Chedeau, and Didier Verna. In *ELS'12, 5th European Lisp Symposium*, Zadar, Croatia, May 2012. DOI [10.5281/zenodo.3248934](https://doi.org/10.5281/zenodo.3248934).
28. Star T_EX: the next generation. Didier Verna. In Barbara Beeton and Karl Berry, editors, *TUG'12, 33rd T_EX Users Group Conference*, volume 33. T_EX Users Group, 2012.
29. 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](https://doi.org/10.1145/2089131.2089140).
30. Towards $\mathbb{E}$ T_EX coding standards. Didier Verna. In Barbara Beeton and Karl Berry, editors, *TUG'11, 32nd T_EX Users Group Conference*, volume 32, pages 309–328. T_EX Users Group, 2011.
31. CLoX: Common lisp objects for XEmacs. Didier Verna. In *ELS'10, 3rd European Lisp Symposium*, Lisbon, Portugal, May 2010. DOI [10.5281/zenodo.3248958](https://doi.org/10.5281/zenodo.3248958).
32. Classes, styles, conflicts: the biological realm of $\mathbb{E}$ T_EX. Didier Verna. In Barbara Beeton and Karl Berry, editors, *TUG'10, 31st T_EX Users Group Conference*, volume 31, pages 162–172. T_EX Users Group, 2010.
33. 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](https://doi.org/10.5281/zenodo.3386206).
34. 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](https://doi.org/10.5281/zenodo.3248977).
35. Action recognition: How intelligent virtual environments can ease human-machine interaction. Didier Verna. In Hal Thwaites and Scott Thrane Refsland, editors, *VSM'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.
36. *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, *VSM'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.
37. The multicast support in XEmacs. Didier Verna. In *m17n'99, 3rd International Symposium on Multilingual Environments*, Tsukuba, Japan, 1999.
38. Ergonomics and human-machine interaction concerns in Mule. Didier Verna. In *m17n'99, 3rd International Symposium on Multilingual Environments*, Tsukuba, Japan, 1999.
39. 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.
40. 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 Articles de Workshops Internationaux

41. 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.

42. Context-oriented image processing. Didier Verna and François Ripault. In *COP'15, Context-Oriented Programming Workshop*, 2015. ISBN 9781450336543. DOI [10.1145/2786545.2786547](https://doi.org/10.1145/2786545.2786547).
43. Beating C in scientific computing applications. Didier Verna. In *ELW'06, 3rd European Lisp Workshop*, Nantes, France, July 2006.
44. 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 Articles de Conférences Francophones

45. 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.
46. 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.
47. 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.
48. Définir le virtuel: une vision cognitive. Didier Verna. In *ReViCo'99, Réalité Virtuelle et Cognition*, Paris, France, December 1999.

5.6 Keynotes et Exposés sur Invitation

49. Traitement des similarités dans la justification de paragraphe. Didier Verna. Exposé GUTenberg, September 2025.
50. Justification de paragraphe: le Knuth-Plass. Didier Verna. Exposé GUTenberg, January 2024.
51. Biological realms in computer science. Didier Verna. Keynote at ACCU'14, apr 2014.
52. CLOS efficiency: Instantiation. Didier Verna. Invited Talk at the Vrije University of Brussels, 2010.
53. Scientific computing in lisp: Beyond the performance of C. Didier Verna. Invited Talk at LaBRI, Université de Bordeaux I, France, 2006.

5.7 Rapports et publications diverses

54. *(Dynamic (Programming Paradigms)) ; Performance and Expressivity*. Didier Verna. PhD thesis, EDITE de Paris, Sorbone-Université, EPITA/LRDE, July 2020. Habilitation à Diriger les Recherches.
55. JSPP: Morphing C++ into JavaScript. Christopher Chedeau and Didier Verna. Technical Report 201201-TR, LRDE (EPITA Research and Development Laboratory), January 2012.
56. Standard output streams default behavior in terminal sessions. Didier Verna. Common Document Repository #11, 2012. DOI [10.5281/zenodo.3414042](https://doi.org/10.5281/zenodo.3414042).
57. Clarification proposal for CLHS 22.3. Didier Verna. Common Document Repository #7, 2011. DOI [10.5281/zenodo.3413913](https://doi.org/10.5281/zenodo.3413913).
58. File-local variables. Didier Verna. Common Document Repository #9, 2011. DOI [10.5281/zenodo.3414042](https://doi.org/10.5281/zenodo.3414042).
59. $\LaTeX$ curricula vitae with the CurVe class. Didier Verna. *The Prac $\TeX$ Journal*, (3), August 2006.
60. CV formatting with CurVe. Didier Verna. *TUGBoat, Communications of the $\TeX$ Users Group*, 22(4):361–364, December 2001. ISSN 0896320.
61. *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.
62. 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 Exposés Oraux

63. A taste of Julia. Didier Verna. ACCU'16, April 2016.
64. A taste of Julia. Didier Verna. Séminaire Performance et Généricité du LRDE, April 2016.
65. La musique des programmes. Didier Verna. Soirée Thématique de l'EPITA: l'Esthétique en Informatique, 2016.
66. Referential transparency is overrated. Didier Verna. ACCU'15, April 2015.
67. The bright side of exceptions. Didier Verna. ACCU'13, April 2013.

68. Extensibility for DSL design and implementation: a case study in Lisp. Didier Verna. DSLDI'13, DSL Design and Implementation Workshop, April 2013.
69. DSLs from the perspective of extensible languages. Didier Verna. ACCU'12, April 2012.
70. Lisp extensibility: Impact on DSL design and implementation. Didier Verna. Tutorial at ILC'12, the International Lisp Conference, 2012.
71. Meta-circularity, and vice-versa. Didier Verna. ACCU'11, April 2011.
72. Clon, the command-line options nuker. Didier Verna. ILC'10, the International Lisp Conference, 2010.
73. Revisiting the visitor: the just do it pattern. Didier Verna. ACCU'09, April 2009.
74. Performance and genericity: the forgotten power of Lisp. Didier Verna. ACCU'08, April 2008.
75. 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.
76. 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.