

# Pierre-Yves Schobbens

## List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/1762039/publications.pdf>

Version: 2024-02-01

99  
papers

2,986  
citations

361413

20  
h-index

302126

39  
g-index

106  
all docs

106  
docs citations

106  
times ranked

1145  
citing authors

#	ARTICLE	IF	CITATIONS
1	Featured Scents: Towards Assessing Architectural Smells for Self-Adaptive Systems at Runtime. , 2022, ,		2
2	InFoCPS: Integrating Formal Analysis of Cyber-Physical Systems with Energy Prognostics. , 2020, ,		1
3	A Decade of Featured Transition Systems. Lecture Notes in Computer Science, 2019, , 285-312.	1.3	10
4	\$\$extsf {ML}_{u}\$\$ : A Distributed Real-Time Modal Logic. Lecture Notes in Computer Science, 2019, , 19-35.	1.3	1
5	Feature interaction in software product line engineering: A systematic mapping study. Information and Software Technology, 2018, 98, 44-58.	4.4	28
6	Model-based mutant equivalence detection using automata language equivalence and simulations. Journal of Systems and Software, 2018, 141, 1-15.	4.5	9
7	Feature-family-based reliability analysis of software product lines. Information and Software Technology, 2018, 94, 59-81.	4.4	10
8	All roads lead to Rome: Commuting strategies for product-line reliability analysis. Science of Computer Programming, 2018, 152, 116-160.	1.9	4
9	Model-Based Mutation Operators for Timed Systems: A Taxonomy and Research Agenda. , 2018, ,		4
10	Statistical prioritization for software product line testing: an experience report. Software and Systems Modeling, 2017, 16, 153-171.	2.7	27
11	On Featured Transition Systems. Lecture Notes in Computer Science, 2017, , 453-463.	1.3	4
12	Automata Language Equivalence vs. Simulations for Model-Based Mutant Equivalence: An Empirical Evaluation. , 2017, ,		2
13	Formal Analysis of Object-Oriented Mograms. , 2017, ,		0
14	Multi-timed Bisimulation for Distributed Timed Automata. Lecture Notes in Computer Science, 2017, , 52-67.	1.3	4
15	Reusable self-adaptation through bidirectional programming. , 2016, ,		3
16	Featured model types. , 2016, ,		12
17	Featured model-based mutation analysis. , 2016, ,		42
18	Search-based Similarity-driven Behavioural SPL Testing. , 2016, ,		25

#	ARTICLE	IF	CITATIONS
19	Covering SPL Behaviour with Sampled Configurations. , 2015, , .		13
20	State machine flattening, a mapping study and tools assessment. , 2015, , .		5
21	Poster: VIBeS, Transition System Mutation Made Easy. , 2015, , .		4
22	Modeling and Verification for Probabilistic Properties in Software Product Lines. , 2015, , .		25
23	Towards statistical prioritization for software product lines testing. , 2014, , .		24
24	Coverage Criteria for Behavioural Testing of Software Product Lines. Lecture Notes in Computer Science, 2014, , 336-350.	1.3	17
25	An extensible platform for product-line behavioural analysis. , 2014, , .		3
26	Counterexample guided abstraction refinement of product-line behavioural models. , 2014, , .		18
27	Abstract test case generation for behavioural testing of software product lines. , 2014, , .		18
28	Schedulability analysis support for automotive systems. , 2014, , .		18
29	A variability perspective of mutation analysis. , 2014, , .		12
30	Formal semantics, modular specification, and symbolic verification of product-line behaviour. Science of Computer Programming, 2014, 80, 416-439.	1.9	40
31	ArThUR: A Tool for Markov Logic Network. Lecture Notes in Computer Science, 2014, , 319-328.	1.3	1
32	Extending EAST-ADL towards formal modeling and analysis of energy-aware real-time systems. , 2013, , .		4
33	Supporting multiple perspectives in feature-based configuration. Software and Systems Modeling, 2013, 12, 641-663.	2.7	37
34	A methodology for formal analysis and verification of EAST-ADL models. Reliability Engineering and System Safety, 2013, 120, 127-138.	8.9	26
35	Model-Based Verification of Energy-Aware Real-Time Automotive Systems. , 2013, , .		10
36	A framework for the rigorous design of highly adaptive timed systems. , 2013, , .		8

#	ARTICLE	IF	CITATIONS
37	Beyond Boolean product-line model checking: Dealing with feature attributes and multi-features. , 2013, , .		37
38	Featured Transition Systems: Foundations for Verifying Variability-Intensive Systems and Their Application to LTL Model Checking. IEEE Transactions on Software Engineering, 2013, 39, 1069-1089.	5.6	198
39	ProVeLines. , 2013, , .		55
40	Model Checking Adaptive Software with Featured Transition Systems. Lecture Notes in Computer Science, 2013, , 1-29.	1.3	19
41	Behavioural modelling and verification of real-time software product lines. , 2012, , .		37
42	Managing evolution in software product lines. , 2012, , .		17
43	Model checking software product lines with SNIP. International Journal on Software Tools for Technology Transfer, 2012, 14, 589-612.	1.9	90
44	Towards Formal Energy and Time Aware Behaviors in EAST-ADL: An MDE Approach. , 2012, , .		8
45	Simulation-based abstractions for software product-line model checking. , 2012, , .		25
46	Towards an incremental automata-based approach for software product-line model checking. , 2012, , .		14
47	A Vision for Behavioural Model-Driven Validation of Software Product Lines. Lecture Notes in Computer Science, 2012, , 208-222.	1.3	14
48	Symbolic model checking of software product lines. , 2011, , .		141
49	Distributed Event Clock Automata. Lecture Notes in Computer Science, 2011, , 250-263.	1.3	4
50	Model checking <u>lots</u> of systems. , 2010, , .		242
51	Memory Event Clocks. Lecture Notes in Computer Science, 2010, , 198-212.	1.3	3
52	Tool support for code generation from a UMLsec property. , 2010, , .		8
53	Towards Multi-view Feature-Based Configuration. Lecture Notes in Computer Science, 2010, , 106-112.	1.3	21
54	Model Co-evolution and Consistency Management (MCCMâ€™08). Lecture Notes in Computer Science, 2009, , 120-123.	1.3	0

#	ARTICLE	IF	CITATIONS
55	Clear justification of modeling decisions for goal-oriented requirements engineering. Requirements Engineering, 2008, 13, 87-115.	3.1	43
56	Evaluating formal properties of feature diagram languages. IET Software, 2008, 2, 281.	2.1	46
57	Whatâ€™s in a Feature: A Requirements Engineering Perspective. , 2008, , 16-30.		74
58	Disambiguating the Documentation of Variability in Software Product Lines: A Separation of Concerns, Formalization and Automated Analysis. , 2007, , .		132
59	Generic semantics of feature diagrams. Computer Networks, 2007, 51, 456-479.	5.1	329
60	The computational complexity of scenario-based agent verification and design. Journal of Applied Logic, 2007, 5, 252-276.	1.1	13
61	Model-checking the preservation of temporal properties upon feature integration. International Journal on Software Tools for Technology Transfer, 2007, 9, 53-62.	1.9	3
62	SynthÃ©se de diagrammes d'Ã©tats par classe Ã partir de diagrammes de sÃ©quence. Techniques Et Sciences Informatiques, 2007, 26, 797-817.	0.0	0
63	Achieving, Satisficing, and Excelling. , 2007, , 286-295.		3
64	Justifying Goal Models. , 2006, , .		7
65	Feature Diagrams: A Survey and a Formal Semantics. , 2006, , .		283
66	Synthesising Features by Games. Electronic Notes in Theoretical Computer Science, 2006, 145, 79-93.	0.9	4
67	A More Expressive Softgoal Conceptualization for Quality Requirements Analysis. Lecture Notes in Computer Science, 2006, , 281-295.	1.3	27
68	Comparative semantics of Feature Diagrams: FFD vs. vDFD. , 2006, , .		3
69	Allocating Goals to Agent Roles During MAS Requirements Engineering. , 2006, , 19-34.		1
70	Model-checking the Preservation of Temporal Properties upon Feature Integration. Electronic Notes in Theoretical Computer Science, 2005, 128, 311-324.	0.9	3
71	Lightweight Formal Methods for Scenario-Based Software Engineering. Lecture Notes in Computer Science, 2005, , 174-192.	1.3	6
72	From live sequence charts to state machines and back: a guided tour. IEEE Transactions on Software Engineering, 2005, 31, 999-1014.	5.6	40

#	ARTICLE	IF	CITATIONS
73	A New Algorithm for Strategy Synthesis in LTL Games. Lecture Notes in Computer Science, 2005, , 477-492.	1.3	23
74	An Algebraic Approach for Codesign. Lecture Notes in Computer Science, 2005, , 415-430.	1.3	0
75	The Complexity of Live Sequence Charts. Lecture Notes in Computer Science, 2005, , 364-378.	1.3	5
76	Alternating-time logic with imperfect recall. Electronic Notes in Theoretical Computer Science, 2004, 85, 82-93.	0.9	99
77	Model-Checking Access Control Policies. Lecture Notes in Computer Science, 2004, , 219-230.	1.3	35
78	Operators and Laws for Combining Preference Relations. Journal of Logic and Computation, 2002, 12, 13-53.	0.8	92
79	Axioms for real-time logics. Theoretical Computer Science, 2002, 274, 151-182.	0.9	19
80	A two-level temporal logic for evolving specifications. Information Processing Letters, 2002, 83, 167-172.	0.6	1
81	Approximating ATL* in ATL. Lecture Notes in Computer Science, 2002, , 289-301.	1.3	7
82	Model-Generation of a Fictitious Clock Real-Time Logic Using Sharing Trees. Electronic Notes in Theoretical Computer Science, 2001, 23, 108-126.	0.9	0
83	Proving feature non-interaction with Alternating-Time Temporal Logic. , 2001, , 85-103.		4
84	The logic of "initially" and "next". Complete axiomatization and complexity. Information Processing Letters, 1999, 69, 221-225.	0.6	4
85	Formal Models of Agents. Lecture Notes in Computer Science, 1999, , .	1.3	9
86	The regular real-time languages. Lecture Notes in Computer Science, 1998, , 580-591.	1.3	67
87	State clock logic: A decidable real-time logic. Lecture Notes in Computer Science, 1997, , 33-47.	1.3	43
88	Counterfactuals and Updates as Inverse Modalities. Journal of Logic, Language and Information, 1997, 6, 123-146.	0.6	27
89	Belief Revision and Verisimilitude. Notre Dame Journal of Formal Logic, 1995, 36, .	0.4	9
90	Second-order proof systems for algebraic specification languages. Lecture Notes in Computer Science, 1994, , 321-336.	1.3	3

#	ARTICLE	IF	CITATIONS
91	Extensions of initial models and their second-order proof systems. Lecture Notes in Computer Science, 1994, , 326-344.	1.3	1
92	Exceptions for algebraic specifications: on the meaning of "but". Science of Computer Programming, 1993, 20, 73-111.	1.9	12
93	A logic for legal hierarchies. , 1993, , .		1
94	An experiment in formal software development. Communications of the ACM, 1991, 34, 62.	4.5	46
95	Applying VDM to large developments. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 1990, 15, 55-58.	0.7	0
96	Two approaches towards the formalisation of VDM. Lecture Notes in Computer Science, 1990, , 370-398.	1.3	3
97	Applying VDM to large developments. , 1990, , .		2
98	An experiment in formal software development: using the B theorem prover on a VDM case study. , 0, , .		3
99	Synthesis of open reactive systems from scenario-based specifications. , 0, , .		14