

Ian J Hayes

List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/8928158/publications.pdf>

Version: 2024-02-01

72
papers

1,044
citations

623734

14
h-index

477307

29
g-index

80
all docs

80
docs citations

80
times ranked

254
citing authors

#	ARTICLE	IF	CITATIONS
1	Laws of programming. Communications of the ACM, 1987, 30, 672-686.	4.5	331
2	A case-study in timed refinement: a mine pump. IEEE Transactions on Software Engineering, 1992, 18, 817-826.	5.6	64
3	Specification directed module testing. IEEE Transactions on Software Engineering, 1986, SE-12, 124-133.	5.6	51
4	A sequential real-time refinement calculus. Acta Informatica, 2001, 37, 385-448.	0.5	35
5	Determining the Specification of a Control System from That of Its Environment. Lecture Notes in Computer Science, 2003, , 154-169.	1.3	33
6	A timeband framework for modelling real-time systems. Real-Time Systems, 2010, 45, 106-142.	1.3	32
7	Using units of measurement in formal specifications. Formal Aspects of Computing, 1995, 7, 329-347.	1.8	30
8	A semantics for Behavior Trees using CSP with specification commands. Science of Computer Programming, 2011, 76, 891-914.	1.9	25
9	VDM and Z: A comparative case study. Formal Aspects of Computing, 1992, 4, 76-99.	1.8	23
10	Generalised rely-guarantee concurrency: an algebraic foundation. Formal Aspects of Computing, 2016, 28, 1057-1078.	1.8	21
11	A Program Refinement Tool. Formal Aspects of Computing, 1998, 10, 97-124.	1.8	18
12	Reasoning about real-time repetitions: terminating and nonterminating. Science of Computer Programming, 2002, 43, 161-192.	1.9	18
13	Balancing expressiveness in formal approaches to concurrency. Formal Aspects of Computing, 2015, 27, 475-497.	1.8	18
14	Deriving Specifications for Systems That Are Connected to the Physical World. , 2007, , 364-390.		18
15	Specification by interface separation. Formal Aspects of Computing, 1995, 7, 430-439.	1.8	15
16	Comparing Degrees of Non-Determinism in Expression Evaluation. Computer Journal, 2013, 56, 741-755.	2.4	15
17	Supporting contexts in program refinement. Science of Computer Programming, 1997, 29, 279-302.	1.9	14
18	Designing a semantic model for a wide-spectrum language with concurrency. Formal Aspects of Computing, 2017, 29, 853-875.	1.8	14

#	ARTICLE	IF	CITATIONS
19	An Algebra of Synchronous Atomic Steps. Lecture Notes in Computer Science, 2016, , 352-369.	1.3	14
20	CSP with Hierarchical State. Lecture Notes in Computer Science, 2009, , 118-135.	1.3	14
21	Unifying Theories of Programming That Distinguish Nontermination and Abort. Lecture Notes in Computer Science, 2010, , 178-194.	1.3	14
22	An Introduction to Real-Time Object-Z. Formal Aspects of Computing, 2002, 13, 128-141.	1.8	13
23	Towards reasoning about teleo-reactive programs for robust real-time systems. , 2008, , .		11
24	A synchronous program algebra: a basis for reasoning about shared-memory and event-based concurrency. Formal Aspects of Computing, 2019, 31, 133-163.	1.8	11
25	Multi-relations in Z. Acta Informatica, 1992, 29, 33-62.	0.5	9
26	Possible values: Exploring a concept for concurrency. Journal of Logical and Algebraic Methods in Programming, 2016, 85, 972-984.	0.5	9
27	Reasoning about Loops in Total and General Correctness. Lecture Notes in Computer Science, 2010, , 62-81.	1.3	9
28	Reasoning about goal-directed real-time teleo-reactive programs. Formal Aspects of Computing, 2014, 26, 563-589.	1.8	8
29	Continuous Action System Refinement. Lecture Notes in Computer Science, 2006, , 316-337.	1.3	8
30	A Guide to Rely/Guarantee Thinking. Lecture Notes in Computer Science, 2018, , 1-38.	1.3	7
31	Termination of Real-Time Programs: Definitely, Definitely Not, or Maybe. Lecture Notes in Computer Science, 2006, , 141-154.	1.3	7
32	Structural operational semantics through context-dependent behaviour. The Journal of Logic and Algebraic Programming, 2011, 80, 392-426.	1.4	6
33	Convolution as a Unifying Concept. ACM Transactions on Computational Logic, 2016, 17, 1-25.	0.9	6
34	Path-Sensitive Data Flow Analysis Simplified. Lecture Notes in Computer Science, 2013, , 415-430.	1.3	6
35	A refinement calculus for logic programs. Theory and Practice of Logic Programming, 2002, 2, 425-460.	1.5	5
36	A theory for execution-time derivation in real-time programs. Theoretical Computer Science, 2005, 346, 3-27.	0.9	5

#	ARTICLE	IF	CITATIONS
37	Algebraic reasoning for probabilistic action systems and while-loops. Acta Informatica, 2008, 45, 321-382.	0.5	5
38	Linking Unifying Theories of Program refinement. Science of Computer Programming, 2013, 78, 2086-2107.	1.9	5
39	Deriving Real-Time Action Systems Controllers from Multiscale System Specifications. Lecture Notes in Computer Science, 2012, , 102-131.	1.3	5
40	Towards an Algebra for Real-Time Programs. Lecture Notes in Computer Science, 2012, , 50-65.	1.3	5
41	Encoding Fairness in a Synchronous Concurrent Program Algebra. Lecture Notes in Computer Science, 2018, , 222-239.	1.3	5
42	Formal Semantics for Program Paths. Electronic Notes in Theoretical Computer Science, 2003, 78, 58-81.	0.9	4
43	Deriving real-time action systems in a sampling logic. Science of Computer Programming, 2013, 78, 2047-2063.	1.9	4
44	Deriving real-time action systems with multiple time bands using algebraic reasoning. Science of Computer Programming, 2014, 85, 137-165.	1.9	4
45	Reasoning Algebraically About Probabilistic Loops. Lecture Notes in Computer Science, 2006, , 380-399.	1.3	4
46	The Real-Time Refinement Calculus: A Foundation for Machine-Independent Real-Time Programming. Lecture Notes in Computer Science, 2002, , 44-58.	1.3	4
47	Programs as Paths: An Approach to Timing Constraint Analysis. Lecture Notes in Computer Science, 2003, , 1-15.	1.3	4
48	A Formal Semantics of the GraalVM Intermediate Representation. Lecture Notes in Computer Science, 2021, , 111-126.	1.3	4
49	Compositional Action System Derivation Using Enforced Properties. Lecture Notes in Computer Science, 2010, , 119-139.	1.3	4
50	Model-Driven Web Form Validation with UML and OCL. Lecture Notes in Computer Science, 2012, , 223-235.	1.3	4
51	Enforcing Safety and Progress Properties: An Approach to Concurrent Program Derivation. , 2009, , .		3
52	Visuocode: A software development environment that supports spatial navigation and composition. , 2013, , .		3
53	Supporting module reuse in refinement. Science of Computer Programming, 1996, 27, 175-184.	1.9	2
54	Using Theory Interpretation to Mechanise the Reals in a Theorem Prover. Electronic Notes in Theoretical Computer Science, 2001, 42, 266-281.	0.9	2

#	ARTICLE	IF	CITATIONS
55	Linear Approximation of Execution-Time Constraints. Formal Aspects of Computing, 2003, 15, 319-348.	1.8	2
56	Developing Logic Programs from Specifications Using Stepwise Refinement. Lecture Notes in Computer Science, 2004, , 66-89.	1.3	2
57	Calculating modules in contextual logic program refinement. Theory and Practice of Logic Programming, 2008, 8, 1-31.	1.5	2
58	Deriving Specifications of Control Programs for Cyber Physical Systems. Computer Journal, 2020, 63, 774-790.	2.4	2
59	Dynamically Detecting Faults via Integrity Constraints. Lecture Notes in Computer Science, 2009, , 85-103.	1.3	2
60	Integrated Operational Semantics: Small-Step, Big-Step and Multi-step. Lecture Notes in Computer Science, 2012, , 21-35.	1.3	2
61	Developing an Algebra for Rely/Guarantee Concurrency: Design Decisions and Challenges. Lecture Notes in Computer Science, 2019, , 176-197.	1.3	2
62	Probabilistic Choice in Refinement Algebra. Lecture Notes in Computer Science, 2008, , 243-267.	1.3	2
63	Expressive Power of Specification Languages. Formal Aspects of Computing, 1998, 10, 187-192.	1.8	1
64	Don't Care Non-determinism in Logic Program Refinement. Electronic Notes in Theoretical Computer Science, 2002, 61, 101-121.	0.9	1
65	Procedures and parameters in the real-time program refinement calculus. Science of Computer Programming, 2007, 64, 286-311.	1.9	1
66	Type Capabilities for Object-Oriented Programming Languages. Lecture Notes in Computer Science, 2018, , 215-230.	1.3	1
67	Invariants and Well-Foundedness in Program Algebra. Lecture Notes in Computer Science, 2010, , 1-14.	1.3	1
68	Engineering a Theory of Concurrent Programming. Lecture Notes in Computer Science, 2018, , 3-18.	1.3	0
69	Software Specification. , 2021, , 251-270.		0
70	Reasoning about Timeouts. Lecture Notes in Computer Science, 2002, , 94-116.	1.3	0
71	Some Challenges of Specifying Concurrent Program Components. Electronic Proceedings in Theoretical Computer Science, EPTCS, 0, 282, 10-22.	0.8	0
72	Some remarks on "Ambiguous Machine Architecture". Computer Architecture News, 1978, 6, 23-24.	2.5	0