

Lars Grunske

List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/6581358/publications.pdf>

Version: 2024-02-01

98
papers

2,900
citations

279487

23
h-index

276539

41
g-index

102
all docs

102
docs citations

102
times ranked

1681
citing authors

#	ARTICLE	IF	CITATIONS
1	Dynamic QoS Management and Optimization in Service-Based Systems. IEEE Transactions on Software Engineering, 2011, 37, 387-409.	4.3	290
2	Software Architecture Optimization Methods: A Systematic Literature Review. IEEE Transactions on Software Engineering, 2013, 39, 658-683.	4.3	221
3	ArcheOpterix: An extendable tool for architecture optimization of AADL models. , 2009, , .		127
4	A learning-to-rank based fault localization approach using likely invariants. , 2016, , .		104
5	Specification patterns for probabilistic quality properties. , 2008, , .		101
6	An Approach to Forecasting QoS Attributes of Web Services Based on ARIMA and GARCH Models. , 2012, , .		98
7	Aligning Qualitative, Real-Time, and Probabilistic Property Specification Patterns Using a Structured English Grammar. IEEE Transactions on Software Engineering, 2015, 41, 620-638.	4.3	88
8	An approach to software reliability prediction based on time series modeling. Journal of Systems and Software, 2013, 86, 1923-1932.	3.3	87
9	Using Models at Runtime to Address Assurance for Self-Adaptive Systems. Lecture Notes in Computer Science, 2014, , 101-136.	1.0	63
10	Model-Driven Safety Evaluation with State-Event-Based Component Failure Annotations. Lecture Notes in Computer Science, 2005, , 33-48.	1.0	53
11	Performance Prediction of Component-Based Systems. Lecture Notes in Computer Science, 2006, , 169-192.	1.0	50
12	An automated approach to forecasting QoS attributes based on linear and non-linear time series modeling. , 2012, , .		49
13	Software Engineering for Self-Adaptive Systems: Research Challenges in the Provision of Assurances. Lecture Notes in Computer Science, 2017, , 3-30.	1.0	49
14	Early quality prediction of component-based systems – A generic framework. Journal of Systems and Software, 2007, 80, 678-686.	3.3	48
15	Quantitative risk-based security prediction for component-based systems with explicitly modeled attack profiles. Journal of Systems and Software, 2008, 81, 1327-1345.	3.3	48
16	Semantic program repair using a reference implementation. , 2018, , .		48
17	Probabilistic Model-Checking Support for FMEA. , 2007, , .		46
18	A Critical Evaluation of Spectrum-Based Fault Localization Techniques on a Large-Scale Software System. , 2017, , .		46

#	ARTICLE	IF	CITATIONS
19	Safety Analysis of an Airbag System Using Probabilistic FMEA and Probabilistic Counterexamples. , 2009, , .		41
20	Dimensions and Metrics for Evaluating Recommendation Systems. , 2014, , 245-273.		41
21	Identifying "good" architectural design alternatives with multi-objective optimization strategies. , 2006, , .		40
22	Hora: Architecture-aware online failure prediction. Journal of Systems and Software, 2018, 137, 669-685.	3.3	39
23	Perpetual Assurances for Self-Adaptive Systems. Lecture Notes in Computer Science, 2017, , 31-63.	1.0	37
24	Reliability-driven deployment optimization for embedded systems. Journal of Systems and Software, 2011, 84, 835-846.	3.3	35
25	Analysing the fitness landscape of search-based software testing problems. Automated Software Engineering, 2017, 24, 603-621.	2.2	35
26	Architecture-driven reliability optimization with uncertain model parameters. Journal of Systems and Software, 2012, 85, 2340-2355.	3.3	34
27	VUDENC: Vulnerability Detection with Deep Learning on a Natural Codebase for Python. Information and Software Technology, 2022, 144, 106809.	3.0	33
28	A Comparative Study into Architecture-Based Safety Evaluation Methodologies Using AADL's Error Annex and Failure Propagation Models. , 2008, , .		32
29	A comparison of machine learning algorithms for proactive hard disk drive failure detection. , 2013, , .		32
30	Experience with fault injection experiments for FMEA. Software - Practice and Experience, 2011, 41, 1233-1258.	2.5	31
31	Monitoring probabilistic properties. , 2009, , .		30
32	Architecture-based reliability evaluation under uncertainty. , 2011, , .		30
33	Test data generation with a Kalman filter-based adaptive genetic algorithm. Journal of Systems and Software, 2015, 103, 343-352.	3.3	30
34	Architecture-Driven Reliability and Energy Optimization for Complex Embedded Systems. Lecture Notes in Computer Science, 2010, , 52-67.	1.0	28
35	Automatic generation of analyzable failure propagation models from component-level failure annotations. , 2005, , .		26
36	Timed Behavior Trees for Failure Mode and Effects Analysis of time-critical systems. Journal of Systems and Software, 2008, 81, 2163-2182.	3.3	26

#	ARTICLE	IF	CITATIONS
37	An evaluation of pure spectrum-based fault localization techniques for large-scale software systems. Software - Practice and Experience, 2019, 49, 1197-1224.	2.5	26
38	Model-based performance analysis of software architectures under uncertainty. , 2013, , .		25
39	Monitoring of Probabilistic Timed Property Sequence Charts. Software - Practice and Experience, 2011, 41, 841-866.	2.5	24
40	Lightweight Adaptive Filtering for Efficient Learning and Updating of Probabilistic Models. , 2015, , .		24
41	Timed Property Sequence Chart. Journal of Systems and Software, 2010, 83, 371-390.	3.3	23
42	HyDiff. , 2020, , .		22
43	Let the Ants Deploy Your Software - An ACO Based Deployment Optimisation Strategy. , 2009, , .		21
44	Formalizing Architectural Refactorings as Graph Transformation Systems. , 0, , .		19
45	Towards an Integration of Standard Component-Based Safety Evaluation Techniques with SaveCCM. Lecture Notes in Computer Science, 2006, , 199-213.	1.0	19
46	How Much Event Data Is Enough? A Statistical Framework for Process Discovery. Lecture Notes in Computer Science, 2018, , 239-256.	1.0	19
47	Mapping the Effectiveness of Automated Test Suite Generation Techniques. IEEE Transactions on Reliability, 2018, 67, 771-785.	3.5	18
48	A Graphical Specification of Model Transformations with Triple Graph Grammars. Lecture Notes in Computer Science, 2005, , 284-298.	1.0	18
49	Timed Behavior Trees and Their Application to Verifying Real-Time Systems. Proceedings / Australian Software Engineering Conference, 2007, , .	0.0	17
50	Defining the abstract syntax of visual languages with advanced graph grammars – A case study based on behavior trees. Journal of Visual Languages and Computing, 2008, 19, 343-379.	1.8	17
51	An effective sequential statistical test for probabilistic monitoring. Information and Software Technology, 2011, 53, 190-199.	3.0	16
52	Evaluating probabilistic models with uncertain model parameters. Software and Systems Modeling, 2014, 13, 1395-1415.	2.2	16
53	Supporting semi-automatic co-evolution of architecture and fault tree models. Journal of Systems and Software, 2018, 142, 115-135.	3.3	16
54	Concolic program repair. , 2021, , .		15

#	ARTICLE	IF	CITATIONS
55	Probabilistic Timed Behavior Trees. Lecture Notes in Computer Science, 2007, , 156-175.	1.0	15
56	An Outline of an Architecture-Based Method for Optimizing Dependability Attributes of Software-Intensive Systems. Lecture Notes in Computer Science, 2007, , 188-209.	1.0	15
57	Specifying model transformations by direct manipulation using concrete visual notations and interactive recommendations. Journal of Visual Languages and Computing, 2015, 28, 195-211.	1.8	14
58	An Architecture-Aware Approach to Hierarchical Online Failure Prediction. , 2016, , .		14
59	History-based Model Repair Recommendations. ACM Transactions on Software Engineering and Methodology, 2021, 30, 1-46.	4.8	14
60	Evolutionary Grammar-Based Fuzzing. Lecture Notes in Computer Science, 2020, , 105-120.	1.0	12
61	Online Workload Forecasting. , 2017, , 529-553.		12
62	BeDivFuzz. , 2022, , .		12
63	An Efficient Method for Architecture-Based Reliability Evaluation for Evolving Systems with Changing Parameters. , 2010, , .		11
64	Choosing the Appropriate Forecasting Model for Predictive Parameter Control. Evolutionary Computation, 2014, 22, 319-349.	2.3	11
65	Inputs From Hell!. IEEE Transactions on Software Engineering, 2022, 48, 1138-1153.	4.3	11
66	A Formal Syntax for Probabilistic Timed Property Sequence Charts. , 2009, , .		10
67	Specification and Evaluation of Safety Properties in a Component-Based Software Engineering Process. Lecture Notes in Computer Science, 2005, , 249-274.	1.0	9
68	An Automated Dependability Analysis Method for COTS-Based Systems. Lecture Notes in Computer Science, 2005, , 178-190.	1.0	8
69	Does Diversity Improve the Test Suite Generation for Mobile Applications?. Lecture Notes in Computer Science, 2019, , 58-74.	1.0	8
70	Using Automated Control Charts for the Runtime Evaluation of QoS Attributes. , 2011, , .		7
71	MoFuzz. , 2020, , .		7
72	PSPWizard. , 2011, , .		6

#	ARTICLE	IF	CITATIONS
73	Statistical detection of QoS violations based on CUSUM control charts. , 2012, , .		6
74	Challenges for Verifying and Validating Scientific Software in Computational Materials Science. , 2019, , .		6
75	Capture and reuse of composable failure patterns. International Journal of Critical Computer-Based Systems, 2010, 1, 128.	0.1	5
76	A systematic literature review on counterexample explanation. Information and Software Technology, 2022, 145, 106800.	3.0	5
77	Generalizable safety annotations for specification of failure patterns. Software - Practice and Experience, 2010, 40, 453-483.	2.5	4
78	Increasing Dependability of Component-Based Software Systems by Online Failure Prediction (Short) Tj ETQq0 0 0 rgBT /Overlock 10 Tf		4
79	CoWolf â€“ A Generic Framework for Multi-view Co-evolution and Evaluation of Models. Lecture Notes in Computer Science, 2015, , 34-40.	1.0	4
80	State Space Reduction Techniques for Component Interfaces. Lecture Notes in Computer Science, 2008, , 130-145.	1.0	4
81	Counterexample Interpretation for Contract-Based Design. Lecture Notes in Computer Science, 2020, , 99-114.	1.0	4
82	A comprehensive empirical evaluation of generating test suites for mobile applications with diversity. Information and Software Technology, 2021, 130, 106436.	3.0	3
83	metaFMEA-A Framework for Reusable FMEAs. Lecture Notes in Computer Science, 2014, , 110-122.	1.0	3
84	Timed Simulation of Extended AADL-Based Architecture Specifications with Timed Abstract State Machines. Lecture Notes in Computer Science, 2009, , 101-115.	1.0	3
85	Behavioral Types for Embedded Software â€“ A Survey. Lecture Notes in Computer Science, 2005, , 82-106.	1.0	3
86	Complete Shadow Symbolic Execution with Java PathFinder. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2019, 44, 15-16.	0.5	3
87	Evaluating Dependability Attributes of Component-Based Specifications. , 2007, , .		2
88	Tool support for automatic model transformation specification using concrete visualisations. , 2013, , .		2
89	CONTINUOUS ASSESSMENT OF DESIGNS AND RE-USE IN MODEL-BASED SAFETY ANALYSIS. IFAC Postprint Volumes IPPV / International Federation of Automatic Control, 2005, 38, 74-79.	0.4	1
90	Guest editorial: advanced topics in automated software engineering. Automated Software Engineering, 2018, 25, 743-744.	2.2	1

#	ARTICLE	IF	CITATIONS
91	Learning from Evolution for Evolution. , 2019, , 255-308.		1
92	Evolutionary algorithms for safety-cost trade-offs in control system design. , 2006, 39, 247-252.		1
93	EVOLUTIONARY ALGORITHMS FOR SAFETY-COST TRADE-OFFS IN CONTROL SYSTEM DESIGN. IFAC Postprint Volumes IPPV / International Federation of Automatic Control, 2006, 39, 249-254.	0.4	0
94	Systems engineering, test and evaluation: maximising customer satisfaction. Innovations in Systems and Software Engineering, 2007, 3, 103-104.	1.6	0
95	First International Workshop on Quantitative Stochastic Models in the Verification and Design of Software Systems (QUOVADIS) / Over	1.0784314	0
96	Quantitative Verification of Stochastic Regular Expressions. Fundamenta Informaticae, 2021, 179, 135-163.	0.3	0
97	Get and Run for Test Case Generation. Lecture Notes in Computer Science, 2020, , 204-219.	1.0	0
98	The Java Pathfinder Workshop 2019. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2020, 45, 20-22.	0.5	0