

Timothy C Lethbridge

List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/2112395/publications.pdf>

Version: 2024-02-01

95
papers

2,331
citations

567281

15
h-index

345221

36
g-index

102
all docs

102
docs citations

102
times ranked

1278
citing authors

#	ARTICLE	IF	CITATIONS
1	Studying Software Engineers: Data Collection Techniques for Software Field Studies. Empirical Software Engineering, 2005, 10, 311-341.	3.9	341
2	How software engineers use documentation: the state of the practice. IEEE Software, 2003, 20, 35-39.	1.8	242
3	What knowledge is important to a software professional?. Computer, 2000, 33, 44-50.	1.1	213
4	The relevance of software documentation, tools and technologies. , 2002, , .		162
5	An examination of software engineering work practices. , 2010, , .		103
6	Improving software practice through education: Challenges and future trends. , 2007, , .		93
7	Summarizing the Content of Large Traces to Facilitate the Understanding of the Behaviour of a Software System. , 0, , .		85
8	Recovering software architecture from the names of source files. Journal of Software: Evolution and Process, 1999, 11, 201-221.	0.4	79
9	Software Engineering Data Collection for Field Studies. , 2008, , 9-34.		59
10	Priorities for the education and training of software engineers. Journal of Systems and Software, 2000, 53, 53-71.	4.5	49
11	Problems and opportunities for model-centric versus code-centric software development. , 2008, , .		48
12	A survey of the relevance of computer science and software engineering education. , 0, , .		47
13	Recovering Behavioral Design Models from Execution Traces. , 0, , .		44
14	The Dagstuhl Middle Metamodel: A Schema For Reverse Engineering. Electronic Notes in Theoretical Computer Science, 2004, 94, 7-18.	0.9	39
15	SE2004: Recommendations for Undergraduate Software Engineering Curricula. IEEE Software, 2006, 23, 19-25.	1.8	38
16	Comparative study of clustering algorithms and abstract representations for software remodularisation. IET Software, 2003, 150, 185.	1.0	34
17	A taxonomy of software types to facilitate search and evidence-based software engineering. , 2008, , .		31
18	Compression techniques to simplify the analysis of large execution traces. , 0, , .		30

#	ARTICLE	IF	CITATIONS
19	CODE4: a unified system for managing conceptual knowledge. International Journal of Human Computer Studies, 1995, 42, 413-451.	5.6	29
20	Supporting software maintenance by mining software update records. , 0, , .		26
21	A Decade of Software Design and Modeling. , 2018, , .		25
22	Teaching UML using umple: Applying model-oriented programming in the classroom. , 2011, , .		24
23	Model-driven rapid prototyping with Umple. Software - Practice and Experience, 2012, 42, 781-797.	3.6	23
24	The relevance of software education: A survey and some recommendations. Annals of Software Engineering, 1998, 6, 91-110.	0.5	21
25	A simple heuristically-based method for expressive Stimulus-Response animation. Computers and Graphics, 1989, 13, 297-303.	2.5	20
26	Student experience with software modeling tools. Software and Systems Modeling, 2019, 18, 3025-3047.	2.7	20
27	A Survey of Tool Use in Modeling Education. , 2017, , .		18
28	Low-Code Is Often High-Code, So We Must Design Low-Code Platforms to Enable Proper Software Engineering. Lecture Notes in Computer Science, 2021, , 202-212.	1.3	17
29	Modeling Practices in Open Source Software. IFIP Advances in Information and Communication Technology, 2013, , 127-139.	0.7	16
30	Umple: Model-driven development for open source and education. Science of Computer Programming, 2021, 208, 102665.	1.9	16
31	Merging Modeling and Programming Using Umple. Lecture Notes in Computer Science, 2016, , 187-197.	1.3	16
32	A Metamodel for Dynamic Information Generated from Object-Oriented Systems. Electronic Notes in Theoretical Computer Science, 2004, 94, 59-69.	0.9	15
33	A metamodel for the compact but lossless exchange of execution traces. Software and Systems Modeling, 2012, 11, 77-98.	2.7	15
34	Umple: A framework for Model Driven Development of Object-Oriented Systems. , 2015, , .		15
35	Umplification: Refactoring to Incrementally Add Abstraction to a Program. , 2010, , .		14
36	Teaching modeling using Umple: Principles for the development of an effective tool. , 2014, , .		14

#	ARTICLE	IF	CITATIONS
37	Metrics for Concept-Oriented Knowledge Bases. International Journal of Software Engineering and Knowledge Engineering, 1998, 08, 161-188.	0.8	12
38	Design and implementation of distributed expert systems: On a control strategy to manage the execution flow of rule activation. Expert Systems With Applications, 2018, 96, 129-148.	7.6	12
39	Improving program comprehension by enhancing program constructs: An analysis of the Umple language. , 2009, , .		11
40	Enhancing program comprehension with recovered state models. , 0, , .		10
41	Requirement traceability: A model-based approach. , 2014, , .		10
42	Adoption of reverse engineering tools: a cognitive perspective and methodology. , 0, , .		9
43	Promoting traits into model-driven development. Software and Systems Modeling, 2017, 16, 997-1017.	2.7	9
44	Improving Code Generation for Associations: Enforcing Multiplicity Constraints and Ensuring Referential Integrity. Studies in Computational Intelligence, 2014, , 129-149.	0.9	9
45	SEAT: a usable trace analysis tool. , 2005, , .		8
46	Guide to the Software Engineering Body of Knowledge (SWEBOK) and the Software Engineering Education Knowledge (SEEK) - a preliminary mapping. , 0, , .		7
47	Temporal Exploration of Software Models: A Tool Feature to Enhance Software Understanding. Reverse Engineering (WCRE), Working Conference on, 2007, , .	0.0	7
48	Umple: Towards combining model driven with prototype driven system development. , 2010, , .		7
49	Combining experiments and grounded theory to evaluate a research prototype: Lessons from the umple model-oriented programming technology. , 2012, , .		7
50	Recovering software architecture from the names of source files. , 1999, 11, 201.		7
51	Exploring a Model-Oriented and Executable Syntax for UML Attributes. Studies in Computational Intelligence, 2014, , 33-53.	0.9	7
52	Generating Software Documentation in Use Case Maps from Filtered Execution Traces. Lecture Notes in Computer Science, 2015, , 177-192.	1.3	7
53	Extracting Document Structure to Facilitate a Knowledge Base Creation for The UML Superstructure Specification. , 2007, , .		6
54	Understanding the complexity embedded in large routine call traces with a focus on program comprehension tasks. IET Software, 2010, 4, 161.	2.1	6

#	ARTICLE	IF	CITATIONS
55	Improving formal analysis of state machines with particular emphasis on and-cross transitions. Computer Languages, Systems and Structures, 2018, 54, 544-585.	1.4	5
56	Evaluating a domain-specialist-oriented knowledge management system. International Journal of Human Computer Studies, 2000, 52, 961-990.	5.6	4
57	Mixing Software Engineering Research and Development--What Needs Ethical Review and What Does Not?. , 2001, 6, 319-321.		4
58	Towards Tracing at the Model Level. , 2012, , .		4
59	Using Umple to Synergistically Process Features, Variants, UML Models and Classic Code. Lecture Notes in Computer Science, 2018, , 69-88.	1.3	4
60	A Novel Approach to Measure Confidence and Uncertainty in Assurance Cases. , 2019, , .		4
61	Adding a Textual Syntax to an Existing Graphical Modeling Language: Experience Report with GRL. Lecture Notes in Computer Science, 2015, , 159-174.	1.3	4
62	Intelligent Search Methods for Software Maintenance. Information Systems Frontiers, 2002, 4, 409-423.	6.4	3
63	Thoughts on software engineering knowledge, and how to organize it. , 0, , .		3
64	A brief summary of cognitive patterns for program comprehension. , 0, , .		3
65	Assessing composition in modeling approaches. , 2012, , .		3
66	A Fully Automated Approach to Discovering Nondeterminism in State Machine Diagrams. , 2016, , .		3
67	The role of semiotic engineering in software engineering. , 2016, , .		3
68	Component-based Modeling in Umple. , 2018, , .		3
69	Umple as a Component-based Language for the Development of Real-time and Embedded Applications. , 2016, , .		3
70	Improvements to the Guide to the Software Engineering Body of Knowledge (SWEBOK) and to the Software Engineering Education Body of Knowledge (SEEK). , 0, , .		2
71	CodeSnippets Plug-in to Eclipse: Introducing Web 2.0 Tagging to Improve Software Developer Recall. , 2007, , .		2
72	Ten Years Later, Experiments with Clustering as a Software Remodularization Method. , 2009, , .		2

#	ARTICLE	IF	CITATIONS
73	Exploring How to Develop Transformations and Tools for Automated Umplification. , 2012, , .		2
74	Model oriented programming: Bridging the code-model divide. , 2013, , .		2
75	The University of Ottawa Undergraduate Software Engineering Program: Leading and Innovative. , 2017, , .		2
76	UmpleOnline as a Testbed for Modeling Empirical Studies: A Position Paper. , 2019, , .		2
77	Umple as a Template Language (Umple-TL). , 2019, , .		2
78	Report from the Dagstuhl seminar on interoperability of reengineering tools. , 0, , .		1
79	IEEE-CS/ACM computing curriculum software engineering volume project. , 0, , .		1
80	Predictive Software Models. , 0, , .		1
81	A study of applying a research prototype tool in industrial practice. , 2010, , .		1
82	Automated Generation of Use Case Descriptions from Problem Frames. , 2010, , .		1
83	Psychophysiological observing and analysis tool for user experience. , 2016, , .		1
84	Optimizing Hierarchical, Concurrent State Machines in Umple for Model Checking. , 2019, , .		1
85	Umple. Advances in Systems Analysis, Software Engineering, and High Performance Computing Book Series, 2021, , 1-25.	0.5	1
86	Umple-TL: A Model-Oriented, Dependency-Free Text Emission Tool. Communications in Computer and Information Science, 2020, , 127-155.	0.5	1
87	Modeling Relevance Relations Using Machine Learning Techniques. , 2007, , 168-207.		1
88	Concurrent Programming using Umple. , 2018, , .		1
89	Umple as a Template Language (Umple-TL). , 2019, , .		1
90	IEEE-CS/ACM computing curricula. SIGCSE Bulletin, 2004, 36, 450-452.	0.1	0

#	ARTICLE	IF	CITATIONS
91	Software Engineering Education (SEE) Research and Publication: Issues, Challenges and Directions. , 2005, , .		0
92	Challenges and opportunities in applying research prototypes and findings into industrial practice. , 2010, , .		0
93	Are our students engaged in their studies?. , 2018, , .		0
94	Summary of the Educatorâ€™s Symposium. Lecture Notes in Computer Science, 2006, , 302-305.	1.3	0
95	Software Engineering 2004 â€” A Jewel in the ACM/IEEE-CS Curricula Effort. , 0, , 417-421.		0