Timothy C Lethbridge

List of Publications by Year
in descending order

95 papers	2,331 citations	643344 15 h-index	388640 36 g-index
102 all docs	102 docs citations	102 times ranked	1438 citing authors

#	Article	IF	Citations
1	Umple. Advances in Systems Analysis, Software Engineering, and High Performance Computing Book Series, 2021, , 1-25.	0.5	1
2	Umple: Model-driven development for open source and education. Science of Computer Programming, 2021, 208, 102665.	1.5	16
3	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.	1.0	17
4	Umple-TL: A Model-Oriented, Dependency-Free Text Emission Tool. Communications in Computer and Information Science, 2020, , 127-155.	0.4	1
5	UmpleOnline as a Testbed for Modeling Empirical Studies: A Position Paper. , 2019, , .		2
6	Optimizing Hierarchical, Concurrent State Machines in Umple for Model Checking. , 2019, , .		1
7	A Novel Approach to Measure Confidence and Uncertainty in Assurance Cases. , 2019, , .		4
8	Student experience with software modeling tools. Software and Systems Modeling, 2019, 18, 3025-3047.	2.2	20
9	Umple as a Template Language (Umple-TL). , 2019, , .		1
10	Umple as a Template Language (Umple-TL). , 2019, , .		2
11	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
12	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.	4.4	12
13	A Decade of Software Design and Modeling. , 2018, , .		25
14	Using Umple to Synergistically Process Features, Variants, UML Models and Classic Code. Lecture Notes in Computer Science, 2018, , 69-88.	1.0	4
15	Are our students engaged in their studies?. , 2018, , .		0
16	Component-based Modeling in Umple. , 2018, , .		3
17	Concurrent Programming using Umple. , 2018, , .		1
18	Promoting traits into model-driven development. Software and Systems Modeling, 2017, 16, 997-1017.	2.2	9

#	Article	IF	Citations
19	A Survey of Tool Use in Modeling Education. , 2017, , .		18
20	The University of Ottawa Undergraduate Software Engineering Program: Leading and Innovative. , 2017, , .		2
21	A Fully Automated Approach to Discovering Nondeterminism in State Machine Diagrams. , 2016, , .		3
22	The role of semiotic engineering in software engineering. , 2016, , .		3
23	Psychophysiological observing and analysis tool for user experience. , 2016, , .		1
24	Merging Modeling and Programming Using Umple. Lecture Notes in Computer Science, 2016, , 187-197.	1.0	16
25	Umple as a Component-based Language for the Development of Real-time and Embedded Applications. , 2016, , .		3
26	Umple: A framework for Model Driven Development of Object-Oriented Systems. , 2015, , .		15
27	Adding a Textual Syntax to an Existing Graphical Modeling Language: Experience Report with GRL. Lecture Notes in Computer Science, 2015, , 159-174.	1.0	4
28	Generating Software Documentation in Use Case Maps from Filtered Execution Traces. Lecture Notes in Computer Science, 2015, , 177-192.	1.0	7
29	Teaching modeling using Umple: Principles for the development of an effective tool. , 2014, , .		14
30	Requirement traceability: A model-based approach. , 2014, , .		10
31	Exploring a Model-Oriented and Executable Syntax for UML Attributes. Studies in Computational Intelligence, 2014, , 33-53.	0.7	7
32	Improving Code Generation for Associations: Enforcing Multiplicity Constraints and Ensuring Referential Integrity. Studies in Computational Intelligence, 2014, , 129-149.	0.7	9
33	Model oriented programming: Bridging the code-model divide. , 2013, , .		2
34	Modeling Practices in Open Source Software. IFIP Advances in Information and Communication Technology, 2013, , 127-139.	0.5	16
35	Exploring How to Develop Transformations and Tools for Automated Umplification. , 2012, , .		2
36	Combining experiments and grounded theory to evaluate a research prototype: Lessons from the umple model-oriented programming technology. , 2012, , .		7

#	Article	IF	Citations
37	Towards Tracing at the Model Level. , 2012, , .		4
38	Assessing composition in modeling approaches. , 2012, , .		3
39	Modelâ€driven rapid prototyping with Umple. Software - Practice and Experience, 2012, 42, 781-797.	2.5	23
40	A metamodel for the compact but lossless exchange of execution traces. Software and Systems Modeling, 2012, 11, 77-98.	2.2	15
41	Teaching UML using umple: Applying model-oriented programming in the classroom. , 2011, , .		24
42	Understanding the complexity embedded in large routine call traces with a focus on program comprehension tasks. IET Software, 2010, 4, 161.	1.5	6
43	Umple: Towards combining model driven with prototype driven system development. , 2010, , .		7
44	A study of applying a research prototype tool in industrial practice. , 2010, , .		1
45	Challenges and opportunities in applying research prototypes and findings into industrial practice. , 2010, , .		0
46	An examination of software engineering work practices. , 2010, , .		103
47	Automated Generation of Use Case Descriptions from Problem Frames. , 2010, , .		1
48	Umplification: Refactoring to Incrementally Add Abstraction to a Program. , 2010, , .		14
49	Improving program comprehension by enhancing program constructs: An analysis of the Umple language. , 2009, , .		11
50	Ten Years Later, Experiments with Clustering as a Software Remodularization Method. , 2009, , .		2
51	A taxonomy of software types to facilitate search and evidence-based software engineering. , 2008, , .		31
52	Problems and opportunities for model-centric versus code-centric software development. , 2008, , .		48
53	Software Engineering Data Collection for Field Studies. , 2008, , 9-34.		59
54	Improving software practice through education: Challenges and future trends. , 2007, , .		93

#	Article	IF	Citations
55	Extracting Document Structure to Facilitate a Knowledge Base Creation for The UML Superstructure Specification. , 2007, , .		6
56	Temporal Exploration of Software Models: A Tool Feature to Enhance Software Understanding. Reverse Engineering (WCRE), Working Conference on, 2007, , .	0.0	7
57	CodeSnippets Plug-in to Eclipse: Introducing Web 2.0 Tagging to Improve Software Developer Recall. , 2007, , .		2
58	Modeling Relevance Relations Using Machine Learning Techniques. , 2007, , 168-207.		1
59	SE2004: Recommendations for Undergraduate Software Engineering Curricula. IEEE Software, 2006, 23, 19-25.	2.1	38
60	Summary of the Educatorâ€™s Symposium. Lecture Notes in Computer Science, 2006, , 302-305.	1.0	0
61	Software Engineering Education (SEE) Research and Publication: Issues, Challenges and Directions. , 2005, , .		0
62	Studying Software Engineers: Data Collection Techniques for Software Field Studies. Empirical Software Engineering, 2005, 10, 311-341.	3.0	341
63	SEAT: a usable trace analysis tool. , 2005, , .		8
64	IEEE-CS/ACM computing curricula. SIGCSE Bulletin, 2004, 36, 450-452.	0.1	0
65	The Dagstuhl Middle Metamodel: A Schema For Reverse Engineering. Electronic Notes in Theoretical Computer Science, 2004, 94, 7-18.	0.9	39
66	A Metamodel for Dynamic Information Generated from Object-Oriented Systems. Electronic Notes in Theoretical Computer Science, 2004, 94, 59-69.	0.9	15
67	How software engineers use documentation: the state of the practice. IEEE Software, 2003, 20, 35-39.	2.1	242
68	Comparative study of clustering algorithms and abstract representations for software remodularisation. IET Software, 2003, 150, 185.	1.0	34
69	The relevance of software documentation, tools and technologies. , 2002, , .		162
70	Intelligent Search Methods for Software Maintenance. Information Systems Frontiers, 2002, 4, 409-423.	4.1	3
71	Mixing Software Engineering Research and Development--What Needs Ethical Review and What Does Not?. , 2001, 6, 319-321.		4
72	Evaluating a domain-specialist-oriented knowledge management system. International Journal of Human Computer Studies, 2000, 52, 961-990.	3.7	4

#	Article	IF	Citations
73	Priorities for the education and training of software engineers. Journal of Systems and Software, 2000, 53, 53-71.	3.3	49
74	What knowledge is important to a software professional?. Computer, 2000, 33, 44-50.	1.2	213
75	Recovering software architecture from the names of source files. Journal of Software: Evolution and Process, 1999, 11, 201-221.	0.5	79
76	Recovering software architecture from the names of source files. , 1999, 11, 201.		7
77	The relevance of software education: A survey and some recommendations. Annals of Software Engineering, 1998, 6, 91-110.	0.5	21
78	Metrics for Concept-Oriented Knowledge Bases. International Journal of Software Engineering and Knowledge Engineering, 1998, 08, 161-188.	0.6	12
79	CODE4: a unified system for managing conceptual knowledge. International Journal of Human Computer Studies, 1995, 42, 413-451.	3.7	29
80	A simple heuristically-based method for expressive Stimulus-Response animation. Computers and Graphics, 1989, 13, 297-303.	1.4	20
81	A survey of the relevance of computer science and software engineering education. , 0, , .		47
82	Adoption of reverse engineering tools: a cognitive perspective and methodology. , 0, , .		9
83	Report from the Dagstuhl seminar on interoperability of reengineering tools. , 0, , .		1
84	Supporting software maintenance by mining software update records. , 0, , .		26
85	Compression techniques to simplify the analysis of large execution traces. , 0, , .		30
86	Enhancing program comprehension with recovered state models. , 0, , .		10
87	Thoughts on software engineering knowledge, and how to organize it. , 0, , .		3
88	IEEE-CS/ACM computing curriculum software engineering volume project. , 0, , .		1
89	Guide to the Software Engineering Body of Knowledge (SWEBOK) and the Software Engineering Education Knowledge (SEEK) - a preliminary mapping. , 0, , .		7
90	Improvements to the Guide to the Software Engineering Body of Knowledge (SWEBOK) and to the Software Engineering Education Body of Knowledge (SEEK). , 0, , .		2

#	Article	IF	Citations
91	Predictive Software Models. , 0, , .		1
92	A brief summary of cognitive patterns for program comprehension. , 0, , .		3
93	Recovering Behavioral Design Models from Execution Traces. , 0, , .		44
94	Summarizing the Content of Large Traces to Facilitate the Understanding of the Behaviour of a Software System. , 0, , .		85
95	Software Engineering 2004 â€" A Jewel in the ACM/IEEE-CS Curricula Effort. , 0, , 417-421.		0