

# Carla Ferreira

## List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/8943265/publications.pdf>

Version: 2024-02-01

33  
papers

677  
citations

933447

10  
h-index

794594

19  
g-index

34  
all docs

34  
docs citations

34  
times ranked

211  
citing authors

#	ARTICLE	IF	CITATIONS
1	A Trace Semantics for Long-Running Transactions. Lecture Notes in Computer Science, 2005, , 133-150.	1.3	96
2	An Operational Semantics for StAC, a Language for Modelling Long-Running Business Transactions. Lecture Notes in Computer Science, 2004, , 87-104.	1.3	92
3	Putting consistency back into eventual consistency. , 2015, , .		78
4	'Cause I'm strong enough: Reasoning about consistency choices in distributed systems. , 2016, , .		77
5	Comparing Two Approaches to Compensable Flow Composition. Lecture Notes in Computer Science, 2005, , 383-397.	1.3	45
6	A Process Compensation Language. Lecture Notes in Computer Science, 2000, , 61-76.	1.3	35
7	Extending the concept of transaction compensation. IBM Systems Journal, 2002, 41, 743-758.	3.0	34
8	On the Expressive Power of Primitives for Compensation Handling. Lecture Notes in Computer Science, 2010, , 366-386.	1.3	26
9	A Process Calculus Analysis of Compensations. Lecture Notes in Computer Science, 2009, , 87-103.	1.3	24
10	The CISE tool. , 2016, , .		22
11	Extending Eventually Consistent Cloud Databases for Enforcing Numeric Invariants. , 2015, , .		20
12	Dynamic Recovering of Long Running Transactions. Lecture Notes in Computer Science, 2009, , 201-215.	1.3	20
13	'Cause I'm strong enough: Reasoning about consistency choices in distributed systems. ACM SIGPLAN Notices, 2016, 51, 371-384.	0.2	19
14	IPA. Proceedings of the VLDB Endowment, 2018, 12, 404-418.	3.8	17
15	Advanced Mechanisms for Service Combination and Transactions. Lecture Notes in Computer Science, 2011, , 302-325.	1.3	8
16	ECROs: building global scale systems from sequential code. , 2021, 5, 1-30.		7
17	OSTRICH - A Type-Safe Template Language for Low-Code Development. , 2021, , .		7
18	Towards Fast Invariant Preservation in Geo-replicated Systems. Operating Systems Review (ACM), 2015, 49, 121-125.	1.9	6

#	ARTICLE	IF	CITATIONS
19	Test mocks for low-code applications built with OutSystems. , 2020, , .		6
20	On the analysis of compensation correctness. The Journal of Logic and Algebraic Programming, 2012, 81, 585-605.	1.4	5
21	Verifying Concurrent Programs Using Contracts. , 2017, , .		5
22	Towards Compensation Correctness in Interactive Systems. Lecture Notes in Computer Science, 2010, , 161-177.	1.3	4
23	Bringing Hybrid Consistency Closer to Programmers. , 2017, , .		3
24	Using SPIN and STeP to Verify Business Processes Specifications. Lecture Notes in Computer Science, 2004, , 207-213.	1.3	3
25	Solution Discovery over Feature Toggling with Built-in Abstraction in OutSystems. , 2021, , .		3
26	The Benefits of Rapid Modelling for E-business System Development. Lecture Notes in Computer Science, 2003, , 17-28.	1.3	2
27	First-Order Dynamic Logic for Compensable Processes. Lecture Notes in Computer Science, 2012, , 104-121.	1.3	2
28	Teaching practical realistic verification of distributed algorithms in Erlang with TLA+. , 2020, , .		2
29	Robust Contract Evolution in a TypeSafe MicroServices Architecture. The Art Science and Engineering of Programming, 2020, 4, .	0.5	2
30	Model-Based Approaches for Validating Business Critical Systems. , 0, , .		1
31	The Case for Fast and Invariant-Preserving Geo-Replication. , 2014, , .		1
32	Making weak consistency great again. , 2016, , .		1
33	Techniques for safe and highly available cloud applications. , 2019, , .		0