Improving Impact of Self-Adaptation and Self-Management Research through Evaluation Methodology

Yuriy Brun

University of Washington

http://www.cs.washington.edu/homes/brun

Statements We Take for Granted

- Self-adaptation and self-management (SASM) are the future.
- SASM research will soon impact everyday software.
- Our work matters to people other than us.

My Goals

- Reasoning why SASM research is important.
- Quantitative analysis of impact SASM research has on premier software engineering venues.
- Increasing that impact through evaluation methodology.

Emergence of New Venues

2004–

ACM Transactions on Autonomous and Adaptive Systems 2006–

SEAMS 2007 Software Engineering for Adaptive and Self-Managing Systems 2006–

2007–

Schloss Dagstuhl Where Computer Scientists Meet 2008, 2010

Industrial Commitment

- Software complexity crisis [IBM01] and autonomic computing [KC03]
- Education: 2 courses at GATech, OSU, and NCSU [KM10]
- NSF Center for Autonomic Computing ($35K annually)
  others involved: UF, UA, Rutgers, BAE Systems, Ball Aerospace, Intel, Microsoft, Northrop-Grumman, NEC, Raytheon, Xerox, Citrix, Imaginestics, and ISCA Technologies

- Large-Scale Internet Computing Initiative [Pre07]
- Over $5 million to study cloud computing [Pre09]

- 2010 appropriations includes adaptive systems technology [Jon09]

Products

"Aspects of autonomic computing have ingrained themselves into products across IBM's entire product line. Rather than becoming a product of its own, autonomic computing improves the quality of existing, and newly developed products."
-Brent A. Miller [KM10], lead architect, IBM

SASM Research in Premier SE Venues

Surveyed seven instances of premier conferences and two years of journals:

- ICSE 2008, 2009, 2010
- FSE 2008, ESEC/FSE 2009
- ASE 2008, 2009
- *TSE* 2008–2009
- *TOSEM* 2008–2009

Survey Results

Venue	SASM papers	Total	Ratio	SASM papers
ICSE 2010	0	52	0.0%	—
ESEC/FSE 2009	2	37	5.4%	[CCG+09, DPT09]
ASE 2009	2	38	5.3%	[BJSS09, WM09]
ICSE 2009	5	50	10.0%	[CK09, CMP09, EGMT09] [MBNJ09, WNGF09]
TSE 2009	0	50	0.0%	—
TOSEM 2009	1	13	7.7%	[MZ09]
FSE 2008	2	31	6.5%	[RSE08, SRWE08]
ASE 2008	1	34	2.9%	[ZSD08]
ICSE 2008	1	55	1.8%	[SBP08]
TSE 2008	4	51	7.8%	[CPS08, CC08, DY08, HOM08]
TOSEM 2008	0	20	0.0%	—
Conference Total	13	297	4.4%	
Journal Total	5	134	3.7%	
Total	18	431	4.2%	

Impact Gap

The survey revealed a gap between the expected impact and the apparent impact of self-adaptation and self-management research.

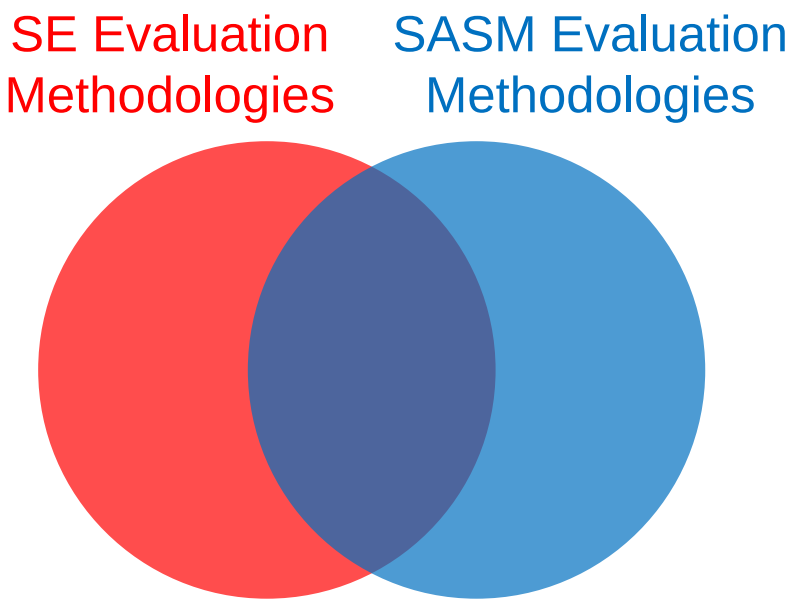
Potentials Reasons for the Impact Gap

- Maturity of the field.
- Emergence of new venues outside of software engineering.
- Relevance to other fields:
  - artificial intelligence
  - pervasive and ubiquitous computing
  - middleware infrastructure
  - multiagent
  - distributed robotics

Why Care about the Gap?

Software engineering can benefit from self-adaptation and self-management research being properly evaluated, reviewed, and published in its venues.

Evaluation Methodology



Many SASM papers involve evaluation methodologies far different from those of SE research.

Evaluation Differences

- Comparison
- Analysis

Evaluation Differences

- Comparison
  - Current technologies cannot handle dynamic environments.
- Analysis

Evaluation Differences

- Comparison
  - Current technologies cannot handle dynamic environments.
  - Certain aspects are crucial to dynamic environments.
- Analysis

Evaluation Differences

- Comparison
  - Current technologies cannot handle dynamic environments.
  - Certain aspects are crucial to dynamic environments.
  - Comparison to naïve SASM solutions.
- Analysis

Evaluation Differences

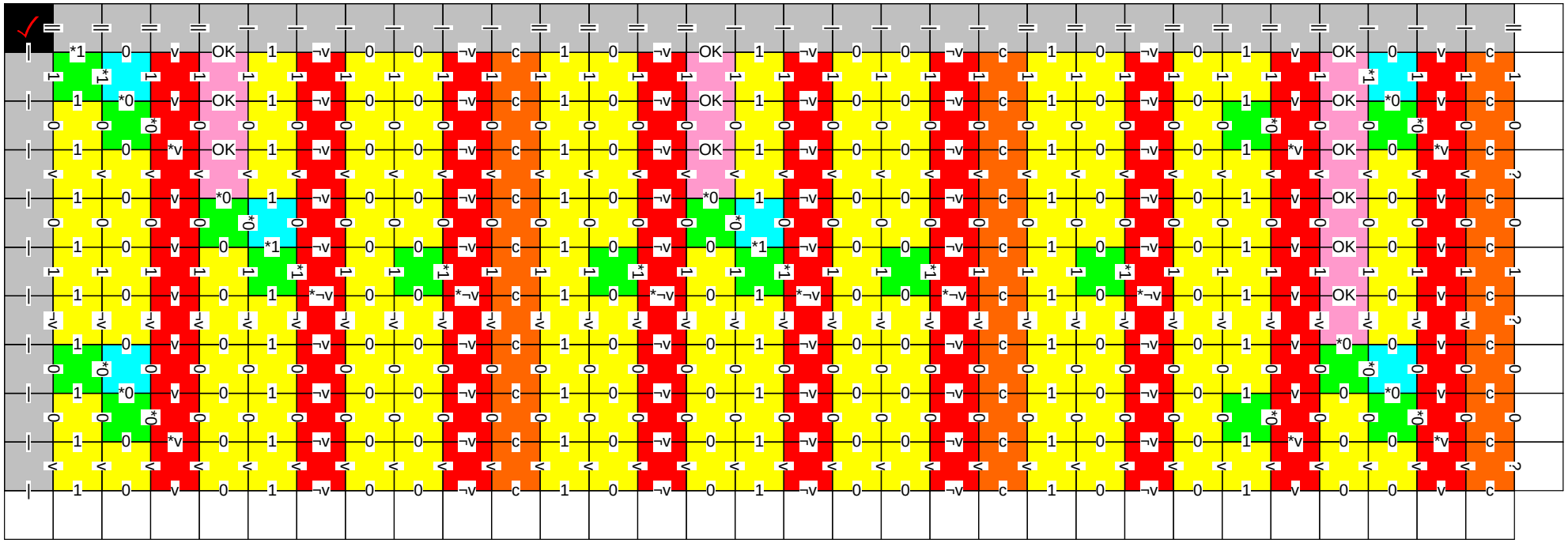
- Comparison
  - Current technologies cannot handle dynamic environments.
  - Certain aspects are crucial to dynamic environments.
  - Comparison to naïve SASM solutions.
  - Comparison to state-of-the-art SASM solutions.
- Analysis

Evaluation Differences

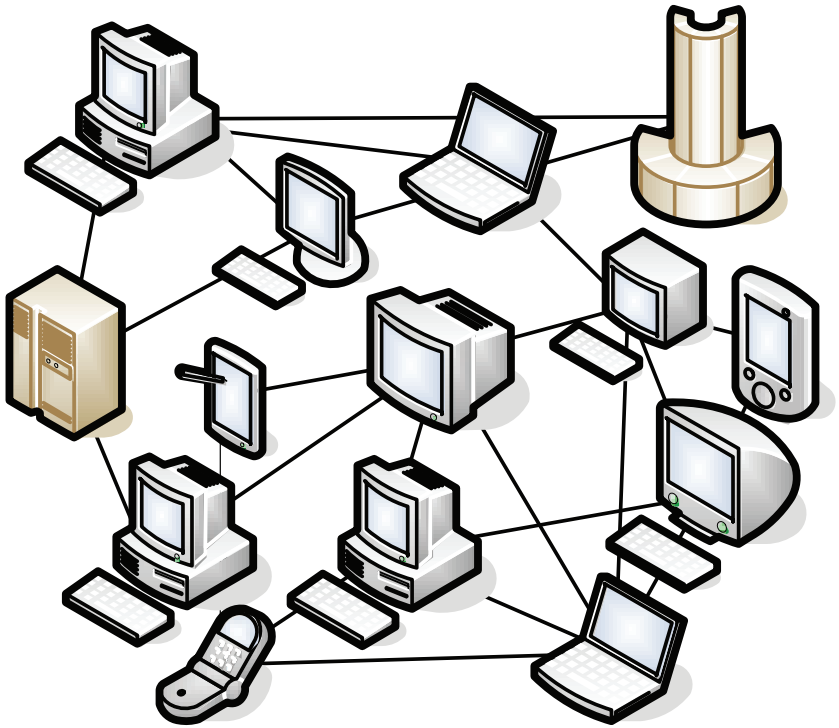
- Comparison
  - Current technologies cannot handle dynamic environments.
  - Certain aspects are crucial to dynamic environments.
  - Comparison to naïve SASM solutions.
  - Comparison to state-of-the-art SASM solutions.
- Analysis
  - Lack of a formal theoretical complexity analysis.

Case Study

Tile architecture: a distributed software system for solving computational problems inspired by nature's self-assembly.



Tile Architecture Approach [BM07a, BM07b]



Distribute a computation so that each machine deals with an insignificant portion of the computation, and no small group of machines knows too much or has too much power.

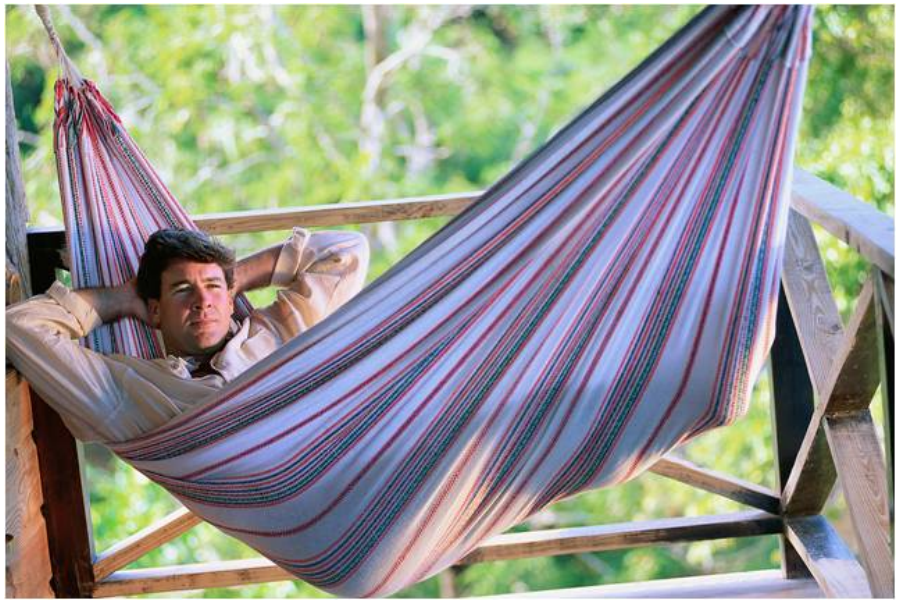
Prototype

1. Build a prototype system.

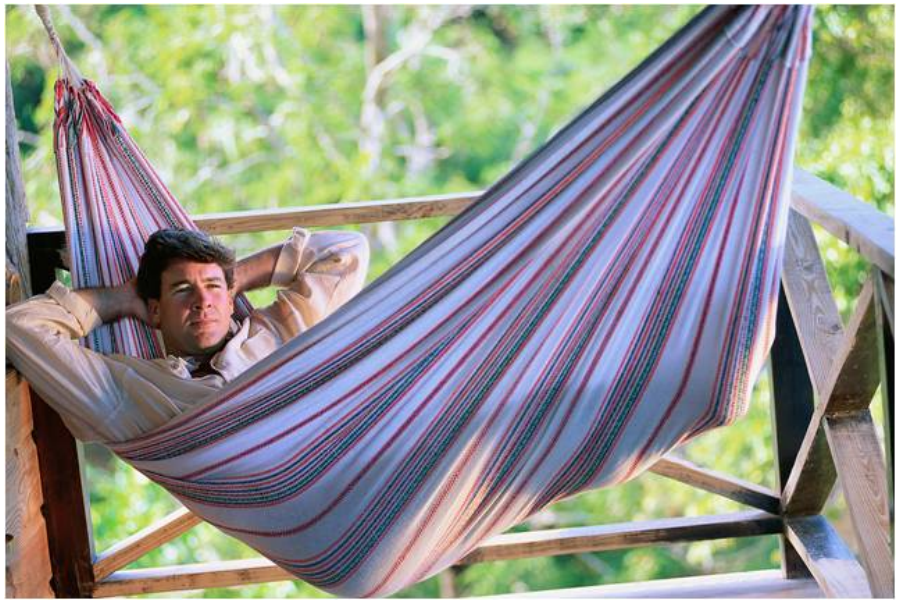
Prototype

1. Build a prototype system.
2. Run it.

Prototype

1. Build a prototype system.
2. Run it.
3. 

Prototype

1. Build a prototype system.
2. Run it.
3. 

- No comparison of any kind!

Comparison

1. Compare running the system on n, $10n$, $100n$, $1000n$ machines.

Comparison

1. Compare running the system on n, $10n$, $100n$, $1000n$ machines.
2. Show scaling trends and great improvement.

Comparison

1. Compare running the system on n, $10n$, $100n$, $1000n$ machines.
2. Show scaling trends and great improvement.
3. 

Comparison

1. Compare running the system on n, $10n$, $100n$, $1000n$ machines.
2. Show scaling trends and great improvement.
3. 

- Self-comparison is fundamentally flawed.

Comparison with State-of-the-Art

1. Theoretical analysis of privacy and security.

Comparison with State-of-the-Art

1. Theoretical analysis of privacy and security.
2. Compare to related work on distributed computation.

Comparison with State-of-the-Art

1. Theoretical analysis of privacy and security.
2. Compare to related work on distributed computation.
3. 

Comparison with State-of-the-Art

1. Theoretical analysis of privacy and security.
2. Compare to related work on distributed computation.
3. 

- Without an analysis of frequency of faults, the reviewer has no way of assessing the benefits.
- Ideas published in a single paper may not be the state-of-the-art.

Analyze the Competition

1. Combine obvious existing solutions.

Analyze the Competition

1. Combine obvious existing solutions.
2. Analyze effects of environment dynamics on existing solutions.

Analyze the Competition

1. Combine obvious existing solutions.
2. Analyze effects of environment dynamics on existing solutions.
3. 

Analyze the Competition

1. Combine obvious existing solutions.
2. Analyze effects of environment dynamics on existing solutions.
3. 

- If your experiments cannot be repeated by independent scientists, you don't get to wear the scientist hat!

Conclusions

- Impact gap: We ♥ SASM research, but it represents only 4.2% of the premier SE papers (none at ICSE 2010).
- Evaluation methodology affects comparing SASM and SE research.
- Careful evaluation can increase the impact of SASM research.

Jacob Burnim, Nicholas Jalbert, Christos Stergiou, and Koushik Sen.
Looper: Lightweight detection of infinite loops at runtime.
In *Proceedings of the 24th IEEE/ACM International Conference on Automated Software Engineering (ASE09)*, Auckland, New Zealand, 2009.

Yuriy Brun and Nenad Medvidovic.
An architectural style for solving computationally intensive problems on large networks.
In *Proceedings of Software Engineering for Adaptive and Self-Managing Systems (SEAMS07)*, Minneapolis, MN, USA, May 2007.

Yuriy Brun and Nenad Medvidovic.
Fault and adversary tolerance as an emergent property of distributed systems' software architectures.
In *Proceedings of the 2nd International Workshop on Engineering Fault Tolerant Systems (EFTS07)*, pages 38–43, Dubrovnik, Croatia, September 2007.

Siu-Nam Chuang and Alvin T.S. Chan.
Dynamic QoS adaptation for mobile middleware.
IEEE Transactions on Software Engineering (TSE), 34(6):738–752, 2008.

Valeria Cardellini, Emiliano Casalicchio, Vincenzo Grassi, Francesco Lo Presti, and Raffaela Mirandola.
QoS-driven runtime adaptation of service oriented architectures.
In *Proceedings of the 7th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE09)*, pages 131–140, Amsterdam, The Netherlands, 2009.

Radu Calinescu and Marta Kwiatkowska.
Using quantitative analysis to implement autonomic IT systems.
In *Proceedings of the ACM/IEEE 31st International Conference on Software Engineering (ICSE09)*, pages 100–110, Vancouver, Canada, 2009.

Hervé Chang, Leonardo Mariani, and Mauro Pezze.
In-field healing of integration problems with COTS components.
In *Proceedings of the ACM/IEEE 31st International Conference on Software Engineering (ICSE09)*, pages 166–176, Vancouver, Canada, 2009.

Carlos Canal, Pascal Poizat, and Gwen Salaün.
Model-based adaptation of behavioral mismatching components.
IEEE Transactions on Software Engineering (TSE), 34(4):546–563, 2008.

Giovanni Denaro, Mauro Pezzè, and Davide Tosi.
Ensuring interoperable service-oriented systems through engineered self-healing.
In *Proceedings of the 7th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE09)*, pages 253–262, Amsterdam, The Netherlands, 2009.

Shlomi Dolev and Reuven Yagel.
Towards self-stabilizing operating systems.
IEEE Transactions on Software Engineering (TSE), 34(4):564–576, 2008.

Ilenia Epifani, Carlo Ghezzi, Raffaela Mirandola, and Giordano Tamburrelli.
Model evolution by run-time parameter adaptation.
In *Proceedings of the ACM/IEEE 31st International Conference on Software Engineering (ICSE09)*, pages 111–121, Vancouver, Canada, 2009.

William G.J. Halfond, Alex Orso, and Pete Manolios.
WASP: Protecting web applications using positive tainting and syntax-aware evaluation.
IEEE Transactions on Software Engineering (TSE), 34:65–81, 2008.

IBM.
Autonomic computing manifesto.

http://www.research.ibm.com/autonomic/manifesto, 2001.

Richard M. Jones.
Fiscal year 2010 National Science Foundation appropriation.
The AIP Bulletin of Science Policy News, (146), 2009.

Jeffrey O. Kephart and David M. Chess.
The vision of autonomic computing.
IEEE Computer, 36(1):41–50, 2003.

Paul Kontogiorgis and Brent A. Miller.
Interview on the state of and IBM's contribution to autonomic computing.
Personal communication, January 2010.

Brice Morin, Olivier Barais, Gregory Nain, and Jean-Marc Jézéquel.
Taming dynamically adaptive systems using models and aspects.
In *Proceedings of the ACM/IEEE 31st International Conference on Software Engineering (ICSE09)*, pages 122–132, Vancouver, Canada, 2009.

Marco Mamei and Franco Zambonelli.
Programming pervasive and mobile computing applications: The TOTA approach.
ACM Transactions on Software Engineering and Methodology (TOSEM), 18(4):1–56, 2009.

Press release.
Google and IBM announce university initiative to address Internet-scale computing challenges.
http://www.google.com/intl/en/press/pressrel/20071008_ibm_univ.html, 2007.

Press release.
National Science Foundation awards millions to fourteen universities for cloud computing research.
http://www.nsf.gov/news/news_summ.jsp?cntn_id=114686&govDel=USNSF_51, 2009.

Franco Raimondi, James Skene, and Wolfgang Emmerich.

Efficient online monitoring of web-service SLAs.
In *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE08)*, pages 170–180, Atlanta, GA, USA, 2008.

Sylvain Sicard, Fabienne Boyer, and Noel De Palma.
Using components for architecture-based management: the self-repair case.
In *Proceedings of the ACM/IEEE 30th International Conference on Software Engineering (ICSE08)*, pages 101–110, Leipzig, Germany, 2008.

Michele Sama, David S. Rosenblum, Zhimin Wang, and Sebastian Elbaum.
Model-based fault detection in context-aware adaptive applications.
In *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE08)*, pages 261–271, Atlanta, GA, USA, 2008.

Yiqiao Wang and John Mylopoulos.

Self-repair through reconfiguration: A requirements engineering approach.
In *Proceedings of the 24th IEEE/ACM International Conference on Automated Software Engineering (ASE09)*, Auckland, New Zealand, 2009.

Westley Weimer, ThanhVu Nguyen, Claire Le Goues, and Stephanie Forrest.
Automatically finding patches using genetic programming.
In *Proceedings of the ACM/IEEE 31st International Conference on Software Engineering (ICSE09)*, pages 364–374, Vancouver, Canada, 2009.

Andrea Zisman, George Spanoudakis, and James Dooley.
A framework for dynamic service discovery.
In *Proceedings of the 23rd IEEE/ACM International Conference on Automated Software Engineering (ASE08)*, pages 158–167, L'Aquila, Italy, 2008.