# Harvey Siy, Ph.D.


Professor
Department of Computer Science
University of Nebraska at Omaha
Omaha, Nebraska 68182

Telephone: (402) 554-2834
Fax: (402) 554-3284
E-mail: hsiy@unomaha.edu
URL: https://hsiy.github.io


## EDUCATION

| | |
|---|---|
| Ph.D. | Computer Science, University of Maryland at College Park, 1996 |
| M.S. | Computer Science, University of Maryland at College Park, 1994 |
| B.S. | Computer Science (*magna cum laude*), University of the Philippines, 1989 |

## PROFESSIONAL EXPERIENCE

| | |
|---|---|
| *2020-* | Professor, Department of Computer Science,<br>University of Nebraska at Omaha |
| *2011-2020* | Associate Professor, Department of Computer Science,<br>University of Nebraska at Omaha |
| *2005-2011* | Assistant Professor, Department of Computer Science,<br>University of Nebraska at Omaha |
| *2003-2005* | Member of Technical Staff, CDMA Wireless Switching Performance and Architecture,<br>Lucent Technologies |
| *1999-2003* | Member of Technical Staff, 5ESS Switching Performance and Architecture,<br>Lucent Technologies |
| *1999* | Member of Technical Staff, Software Technology Transfer Group,<br>Lucent Technologies |
| *1996-1999* | Member of Technical Staff, Software Production Research Department,<br>Bell Labs - Lucent Technologies |
| *1994* | Summer Intern, AT&T Bell Laboratories |
| *1992-1996* | Research Assistant, Department of Computer Science,<br>University of Maryland at College Park |
| *1991-1992* | Instructor, Department of Computer Science,<br>University of the Philippines |
| *1989-1991* | Programmer, Software Brewers Inc., Philippines |

## RESEARCH INTERESTS

I am broadly interested in understanding and solving technical problems related to developing software applications, from embedded software-controlled devices and mobile applications to large software-intensive systems. My objective is to study and improve software engineering techniques that boost the productivity of software developers while preserving the quality (performance, security, reliability, usability, maintainability) of the resulting software products. My research is driven by empirically-validated observations and professional experience with large real-world systems. Current research areas are in software evolution, product line engineering, mining software repositories, empirical methods and computing education.

**JOURNAL PAPERS**

1. Avande, M., Gandhi, R. and Siy, H. (2020). Understanding user engagement with multi-representational license comprehension interfaces. International Journal of Open Source Software and Processes. Online December 14, 2020.
2. Mandal, S., Gandhi, R. and Siy, H. (2020). Modular norm models: practical representation and analysis of contractual rights and obligations. *Requirements Engineering*, 25(3), September 2020, pp. 383-412. (Journal Impact Factor: 2.761.)
3. Robin Gandhi, Keesha Crosby, Harvey Siy and Sayonnha Mandal. (2016). Driving Secure Software Initiatives Using FISMA: Issues and Opportunities. *Crosstalk: The Journal of Defense Software Engineering*, January/February 2016.
4. R. Yokomori, H. Siy, N. Yoshida, M. Noro, K. Inoue. (2015). Further Considerations about Relationship between Framework and Application Components. *International Journal of Computer Science and Application (IJCSA)*, 4(1), April 2015, pp. 18-31.
5. Philip Meyer, Sanjukta Bhowmick and Harvey Siy. (2015). Identifying Important Classes of Large Software Systems Through K-core Decomposition. *Advances in Complex Systems (ACS)*, 17(07n08), Dec. 2014.
6. Robin Gandhi, Keesha Crosby and Harvey Siy. (2014). Gauging the Impact of FISMA on Software Security. *IEEE Computer*, 47(9), Sept. 2014, pp. 103-107.
7. Mariam Rahmani, Azad Azadmanesh, and Harvey Siy. (2014). Architectural Reliability Analysis of Framework-intensive Applications: A Web Case Study. *Journal of Systems and Software (JSS)*, vol. 94, April 2014, pp. 186-201.
8. Yan Wu, Robin Gandhi, and Harvey Siy. (2013). Semi-Automatic Annotation of Natural Language Vulnerability Reports. *International Journal of Secure Software Engineering (IJSSE),* 4(3), 2013, pp. 18-41. *(IGI Global Annual Excellence in Research Journal Award - 2013 Outstanding Article)*
9. R. Yokomori, H. Siy, N. Yoshida, M. Noro, K. Inoue. (2012). Evolution of Component Relationships between Framework and Application, *Journal of Computing*, vol. 23 (2), 2012.
10. V. Winter, J. Perry, H. Siy, S. Srinivasan, B. Farkas, J. McCoy. (2011). The Tyranny of the Vital Few: The Pareto Principle in Language Design. *Journal of Software Engineering and Applications*, Nov. 2011, pp. 146-155.
11. M. deFreitas, H. Siy, M. Zand. (2011). Applying Environmental Factors to Trust Algorithms in Competitive Multi-Agent Systems. *Journal of Software*, 6(11), Nov. 2011, pp. 2263-2270
12. R. Gandhi, H. Siy, Y. Wu. (2010). Studying Software Vulnerabilities. *Crosstalk: The Journal of Defense Software Engineering*, September/October 2010.
13. V. L. Winter, G. Kniesel, H. Siy, M. Zand. (2009). Making Aspect-Orientation Accessible through Syntax-based Language Composition. *IET Software Journal*, 3(3), June 2009, pp. 219-237.
14. H. Siy, P. Chundi, D. Rosenkrantz, M. Subramaniam. (2008). A Segmentation-Based Approach for Temporal Analysis of Software Version Repositories. *Journal on Software Evolution and Maintenance: Research and Practice*, 20(3), May/June 2008, pp. 199-222.
15. Dewayne Perry, Harvey Siy, Lawrence Votta. (2001). Parallel Changes in Large Scale Software Development: An Observational Case Study. *ACM Transactions on Software Engineering and Methodology*, July 2001, pp. 308-377.
16. Todd Graves, Alan Karr, J. Steve Marron, Harvey Siy. (2000). Predicting Fault Incidence Using Software Change History. *IEEE Transactions on Software Engineering*, July 2000, pp. 653-661.

17. Mark Ardis, Nigel Daley, David M. Hoffman, Harvey Siy, and David Weiss. (2000). Software Product Lines: A Case Study. *Software-Practice and Experience*, 30(7):825-847, May 2000.
18. David Atkins, Audris Mockus, Harvey Siy. (2000). Measuring Technology Effects on Software Change Cost. *Bell Labs Technical Journal*, April 2000, pp. 7-18.
19. Harvey Siy, Adam Porter, Lawrence Votta, Audris Mockus. (1998). Understanding the Sources of Variation in Software Inspections. *ACM Transactions on Software Engineering and Methodology*, January 1998, pp. 41-79.
20. Adam Porter, Harvey Siy, Lawrence Votta, Carol Toman. (1997). An Experiment to Assess the Cost-Benefits of Code Inspection in Large Scale Software Development. *IEEE Transactions on Software Engineering*, June 1997, pp. 326-349.

**CONFERENCE PAPERS**

1. Singleton, L., Zhao, R., Song, M. and Siy, H. (2019). CryptoTutor: Teaching Secure Coding through Misuse Pattern Detection, In *Proceedings of the 21st Annual Conference on Information Technology Education (SIGITE 2020)*, Oct. 2020, 6 pages.
2. Orn, D., Duan, L., Liang, Y., Siy, H., Subramaniam, M. (2020). Agro-AI Education: Artificial Intelligence for Future Farmers, In *Proceedings of the 21st Annual Conference on Information Technology Education (SIGITE 2020)*, Oct. 2020, 4 pages.
3. Krutz, J., Siy, H., Dorn, B., Morrison, B. (2019). Stepwise Refinement in Block-Based Programming, In *Consortium for Computing Sciences in Colleges: Midwest (CCSC:MW 2019)*, Oct. 2019, 6 pages.
4. Singleton, L., Zhao, R., Song, M. and Siy, H. (2019). FireBugs: Finding and repairing bugs with security patterns. In *Proceedings of the 6th International Conference on Mobile Software Engineering and Systems (MobileSoft 2019)*, May 2019, pp. 30-34.
5. Sayonnha Mandal, Robin Gandhi, and Harvey Siy. (2017). Modular Norm Models: A Lightweight Approach for Modeling and Reasoning about Legal Compliance. *International Conference on Dependable, Autonomic and Secure Computing (DASC '17)*, Nov. 2017 (to appear).
6. Harvey Siy, Brian Dorn, Carol Engelmann, Neal Grandgenett, Tracie Reding, Jong-Hoon Youn, Qiuming Zhu. (2017). SPARCS: A Personalized Problem-based Learning Approach for Developing Successful Computer Science Learning Experiences in Middle School, *IEEE International Conference on Electro Information Technology (EIT '17)*, Lincoln, Nebraska.
7. Qiuming Zhu, Harvey Siy, Jong-Hoon Youn, Neal Grandgenett and Elliott Ostler. (2014). Enhancing CS Education in High School STEM Curricula: Transferring Research Experience to Classrooms through Problem-Based Learning and Personalized Implementation Plans. *International Conference on Computer Science Education: Innovation & Technology (CSEIT 2014)*, Singapore, September 22-23, 2014, pp. 55-63.
8. Hai-Feng Guo, Zhongyan Qiu, Harvey Siy. (2014). Locating Fault-inducing Patterns from Structural Inputs. *Symposium on Applied Computing (SAC '14), Software Engineering track*, Gyeongju, Korea, pp. 1100-1107, March 2014.
9. Bettina Lechner, Ann Fruhling, Stacie Petter, Harvey Siy. (2013). The Chicken and the Pig: User Involvement in Developing Usability Heuristics. *Americas Conference on Information Systems (AMCIS '13)*, Chicago, IL, August 2013.
10. Ryan Stejskal and Harvey Siy. (2013). Test-Driven Learning in High School Computer Science. *International Conference on Software Engineering Education and Training (CSEE&T '13)*, San Francisco, CA, May 2013.

11. Rahmani, M., Azadmanesh, A., Siy, H. (2012). Petri Net Modeling of Application Server Performance for Web Services, *Int'l Conference on Software Engineering and Knowledge Engineering (SEKE '12)*, Redwood City, CA, July 2012.
12. Paymal, P., Bhowmick, S., Siy, H. (2011). Measuring Disruption From Software Evolution Activities Using Graph-Based Metrics. *Proceedings of the International Conference on Software Maintenance (ICSM) (ERA Track)*, Williamsburg, VA, September, 2011.
13. Y. Wu, H. Siy, R. Gandhi. (2011). Empirical Results from the Study of Software Vulnerabilities (NIER Paper). *New Ideas and Emerging Results Track, International Conference on Software Engineering (ICSE '11)*, Honolulu, Hawaii, May 2011.
14. R. Yokomori, H. Siy, N. Yoshida, M. Noro, K. Inoue. (2011). Measuring the Effects of Aspect-Oriented Refactoring on Component Relationships: Two Case Studies. *International Conference on Aspect-Oriented Software Development (AOSD '11)*, Pernambuco, Brazil, Mar. 2011.
15. R. Yokomori, H. Siy, M. Noro, K. Inoue. (2009). Assessing the Impact of Framework Changes Using Component Ranking. *Proceedings of the International Conference on Software Maintenance (ICSM '09)*, Edmonton, Canada, Sept. 2009.
16. Y. Wu, H. Siy, L. Fan. (2008). Discovering Meaningful Clusters from Mining Software Engineering Literature. *Proceedings of the 20th International Conference on Software Engineering and Knowledge Engineering (SEKE '08)*, Redwood City, California, July 2008.
17. H. Siy, P. Chundi, M. Subramaniam. (2008). Summarizing Developer Histories Using Time-Series Segmentation. *5th Working Conference on Mining Software Repositories (MSR '08)*, Leipzig, Germany, May 2008.
18. H. Siy, P. Chundi, D. Rosenkrantz, M. Subramaniam. (2007). Discovering Dynamic Developer Relationships from Software Version Histories by Time Series Segmentation. *Proceedings of the International Conference on Software Maintenance (ICSM '07)*, Paris, France, Oct. 2007.
19. M. Subramaniam, H. Siy. (2007). Consistently Incorporating Changes to Evolve Transition-based Systems. *Proceedings of the European Conference on Software Maintenance and Reengineering (CSMR '07)*, Amsterdam, Netherlands, Mar. 2007.
20. Y. Wu, H. Siy, M. Zand and V. Winter. (2007). Construction of Ontology-Based Software Repositories by Text Mining. *International Conference on Computational Science (ICCS)*, Beijing, China, May 2007.
21. Harvey Siy, Lawrence Votta. (2001). Does the modern code inspection have value? *Proceedings of the 2001 International Conference on Software Maintenance*, Florence, Italy, Nov. 2001.
22. James Herbsleb, Audris Mockus, Harvey Siy, Mayuram Krishnan, and George Tucker. (2001). Making the Software Factory Work: Lessons from a Decade of Experience. *Proceedings of the 8th International Metrics Symposium (METRICS'01)*, London, U.K., April 2001.
23. Harvey Siy, Audris Mockus. (1999). Measuring Domain Engineering Effects on Software Coding Cost. *Proceedings of the 6th International Metrics Symposium (METRICS'99)*, Boca Raton, Florida, Nov. 1999.
24. N. Staudenmayer, T. Graves, J.S. Marron, A. Mockus, H. Siy, L. Votta, D. Perry. (1998). Adapting to New Environments: How a Legacy Software Organization Copes With Volatility and Change. *5th International Product Development Management Conference*, Como, Italy, May 1998.
25. Dewayne Perry, Harvey Siy, Lawrence Votta. (1998). Parallel Changes in Large Scale Software Development: An Observational Case Study. *Proceedings of the 1998 International Conference on Software Engineering (ICSE'98)*, Kyoto, Japan, April 1998.

26. Adam Porter, Harvey Siy, Lawrence Votta. (1997). Understanding the Effects of Developer Activities on Inspection Interval. *Proceedings of the 1997 International Conference on Software Engineering (ICSE'97)*, Boston, MA, May 1997.
27. Patricia McCarthy, Adam Porter, Harvey Siy, Lawrence Votta. (1996). An Experiment to Assess Cost-Benefits of Inspection Meetings and Their Alternatives. *Proceedings of the Third International Metrics Symposium (METRICS'96)*, Berlin, Germany, Mar. 1996.
28. Adam Porter, Harvey Siy, Carol Toman, Lawrence Votta. (1995). An Experiment to Assess the Cost-Benefits of Code Inspection in Large Scale Software Development. *Proceedings of the Third ACM SIGSOFT Symposium on Foundations of Software Engineering (FSE'95)*, Washington, D.C., Oct. 1995.

**BOOK CHAPTERS**

1. Winter, V., Siy, H., McCoy, J., Farkas, B., Wickstrom, G., Deming, D., Perry, J., Srinivasan, S. (2011). Incorporating Standard Java Libraries into the Design of Embedded Systems. *Java in Academia and Research*, K. Cai, Ed., iConcept Press, 2011.
2. David Atkins, Audris Mockus, Harvey Siy. (2000). Quantifying the Value of New Technologies for Software Development. In *Value-Based Software Engineering*, Eds. Biffl et al., Springer-Verlag, 2006, pp. 327-344.
3. Adam Porter, Harvey Siy, Lawrence Votta. (1996). A Review of Software Inspections. In Marvin Zelkowitz, editor, *Software Process, volume 42 of Advances in Computers*, May 1996.

**WORKSHOP PAPERS**

1. Tracie Reding, Brian Dorn, Harvey Siy, Neal Grandgenett, Jong-Hoon Youn, Qiuming Zhu, Carol Engelmann. (2016). Identification of the Emergent Teacher Leaders within a CSE Professional Development Program, *11th Workshop in Primary and Secondary Computing Education (WiPSCE '16)*, Münster, Germany, pp. 37-44.
2. Sayonnha Mandal, Robin Gandhi, and Harvey Siy. (2016). Can I Copy this Code? Extracting Norms from Software Licenses using Frame Semantics. *Workshop on the Naturalness of Software (NL+SE)*, Nov. 2016.
3. Sayonnha Mandal, Robin Gandhi, and Harvey Siy. (2015). Semantic Web Representations for Reasoning about Applicability and Satisfiability of Federal Regulations for Information Security. *International Workshop on Requirements Engineering and Law (RELAW '15)*, August 2015. *(Best Paper Award.)*
4. Guo, Hai-Feng, Qing Ouyang, and Harvey Siy. (2015). Semantics-based Automated Web Testing. *International Workshop on Automated Specification and Verification of Web Systems (WWV '15)*, June 2015.
5. Gandhi, R.A., Siy, H., Yan, Wu. (2013). Lightweight formal models of software weaknesses. *ICSE International Workshop on Formal Methods in Software Engineering (FormaliSE '13)*, May 2013.
6. McGrath, Scott, Kiran Bastola, and Harvey Siy. (2013). Concept to commit: A pattern designed to trace code changes from user requests to change implementation by analyzing mailing lists and code repositories. *ICSE International Workshop on Data Analysis Patterns in Software Engineering (DAPSE '13)*, May 2013.
7. Dhawal Verma, Jon Gessel, Mansour Zand, Harvey Siy. (2013). Lack of Software Engineering Practices in the Development of Bioinformatics Software. *Eighth International Multi-Conference on Computing in the Global Information Technology (ICCGI 2013)*, July 2013.

8. C. Rahmani, A. Azadmanesh, H. Siy. (2011). Architecture-based Reliability Analysis of Web Services in Multilayer Environment. *ICSE International Workshop on Principles of Engineering Service-Oriented Systems (PESOS '11)*, May 2011.
9. H. Siy, A. Wolfson, M. Zand. (2011). Ontology-Based Product Line Modeling and Generation. *ICSE International Workshop on Product LinE Approaches in Software Engineering (PLEASE '11)*, May 2011.
10. M. deFreitas, M. Zand, H. Siy. (2010). Evaluation of Trust Algorithms in Competitive Multi-Agent Systems. *Annual International Conference on Software Engineering (SE '10)*, Phuket, Thailand, Dec., 2010.
11. C. Rahmani, A. Azadmanesh, H. Siy. (2010). Architecture-based Reliability Modeling of Web Services Using Petri Nets. (Fast Abstract). *International High Assurance Systems Engineering Symposium (HASE '10)*, San Jose, California, Nov., 2010.
12. Y. Wu, R. Gandhi, H. Siy. (2010). Using Semantic Templates to Study Vulnerabilities Recorded in Large Software Repositories. *Proceedings of the ICSE Workshop on Software Engineering for Secure Systems (SESS '10)*, Cape Town, South Africa, May 2010.
13. I. Thapa, H. Siy. (2010). Assessing the Impact of Refactoring Activities on the JHotDraw Project. *Proceedings of the ACM Symposium on Applied Computing (SAC '10)*, Mar. 2010.
14. C. Rahmani, H. Siy, A. Azadmanesh. (2009). An Experimental Analysis of Open Source Software Reliability. *Proceedings of the Field Failure Data Analysis Workshop (F2DA) in Symposium on Reliable Distributed Software Systems (SRDS '09)*, Buffalo, New York, Sept. 2009.
15. C. Rahmani, M. Zand, H. Siy, S. Srinivasan. (2009). A Survey on Model Driven Software Development. *Proceedings of 18th International Conference on Software Engineering and Data Engineering (SEDE '09)*, Las Vegas, Nevada, June 2009.
16. H. Siy, Y. Wu. (2009). An Ontology to Support Empirical Studies in Software Engineering. *Proceedings of the International Conference on Computing in Engineering, Science and Informatics (ICC2009)*, Fullerton, California, April 2009.
17. H. Siy. (2007). Monitoring Offshored/Outsourced Software Maintenance Projects. *Workshop on Accountability and Traceability in Global Software Engineering (ATGSE '07)*.
18. W. L. Sousan, V. L. Winter, M. Zand, and H. Siy. (2007). ERTSAL: A Prototype of a Domain-Specific Aspect Language for Analysis for Embedded Real-Time Systems. *2nd Workshop on Domain Specific Aspect Languages (DSAL)*, Vancouver, British Columbia, March 2007.
19. M. Subramaniam, P. Chundi, H. Siy. (2007). Aggregating Changes to Efficiently Check Consistency. *Proceedings of the International Workshop on Principles of Software Evolution (IWPSE '07)*, Dubrovnik, Croatia, Sept. 2007.
20. H. Siy, P.R. Aryal, V. L. Winter, and M. Zand. (2007). Aspectual Support for Specifying Requirements in Software Product Lines. *Proceedings of Early Aspects at ICSE: Workshop in Aspect-Oriented Requirements Engineering and Architecture Design*, Minneapolis, Minnesota, May 2007.
21. Yan Wu, Mansour Zand, Victor Winter, Harvey Siy. (2006). Systematic text-mining approach for deriving aspects and patterns from domain knowledge. *ICSE 2006 Workshop on Early Aspects*, Shanghai, China, May 2006.
22. Victor Winter, Harvey Siy, Mansour Zand, Prasanna Aryal. (2006). Aspect Traceability through Invertible Weaving. *AOSD 2006 Early Aspects Workshop on Traceability*, Bonn, Germany, Mar. 2006.

23. Harvey Siy, Mansour Zand, Victor Winter. (2005). The Role of Aspects in Domain Engineering. *SPLC 2005 Workshop on Aspects and Product Lines*, Rennes, France, Sept. 2005.
24. Harvey Siy, Dewayne Perry. (1998). Challenges in Large Scale Software Evolution. *ICSE 1998 International Workshop on Principles of Software Evolution (IWPSE)*, Kyoto, Japan, April 1998.
25. Thomas Ball, Yung-Min Kim, Adam Porter, Harvey Siy. (1997). If Your Version Control System Could Talk... *ICSE 1997 Workshop on Process Modeling and Empirical Studies of Software Engineering*, Boston, Massachusetts, May 1997.
26. Adam Porter, Harvey Siy, Carol Toman, Lawrence Votta. (1994). An Experiment to Assess the Cost-Benefits of Code Inspection in Large Scale Software Development: A Preliminary Report. *19th Annual Software Engineering Workshop (SEW)*, NASA Goddard Space Flight Center, Greenbelt, Maryland, Nov. 1994.

## EXTERNAL GRANTS

1. PI. Innovative Technology Experiences for Students and Teachers (ITEST): Strategic Problem-based Approach to Rouse Computer Science (SPARCS), NSF, $1,114,424, (2015-2017).
2. PI. Source Code Analysis to Support Software Modernization. First Data Corporation, $75,000, (2014-2015).
3. PI. Getting Computer Science into the K-12 Curriculum. Google CS4HS, $12,500, (2013).
4. PI. Getting Computer Science into the K-12 Curriculum. Google CS4HS, $13,120, (2012).
5. PI. Identifying Reusable Platforms Through Ontology-Driven Product Line Analysis. NSF EPSCoR University-Industry R&D Partnership, $20,000, (2009-2010).
6. PI. Developer Collaboration on the Jazz Platform: An Empirical Study. IBM Jazz Innovation Award, $25,000, (2008-2009).
7. Co-PI. Safety Software Assurance Compliance and Risk Evaluation (S-SACRE). DoD NAVAIR/Tri-Guard Solutions, $139,468, (2019-2020).
8. Co-PI. Software Assurance Compliance and Risk Evaluation (SACRE) Enhancements. Tri-Guard Solutions, $199,939, (2018-2019).
9. Co-PI. Computer Science Education Roadshow – Opportunities and Pathways. Google CS4MS, $35,000, (2017-2018).
10. Co-PI. Preparing Teachers for Computer Science Principles. Google CS4HS, $35,000, (2016-2017).
11. Co-PI. An Engagement to Look Forward to: Security Requirements and Software Weaknesses. DHS, $649,614, (2013-2014).
12. Co-PI. Developing Precise and Accurate Descriptions of Common Software Weaknesses. NIST, $245,883, (2012-2014).
13. Co-PI. Research Experiences for Teachers (RET) in Engineering and Computer Science: Infusing Mobile Platform Applied Research into Teaching (IMPART), NSF, $449,775, (2012-2015).
14. Co-PI. Evolutionary Adaptation of Java Libraries. Sandia, $240,000, (2010).
15. Senior Personnel. Research Experiences for Teachers (RET) in Engineering and Computer Science: Wearable Research for In-Service STEM Teachers (WRIST), NSF, $598,539, (2017-2020).

16. Senior Personnel. SBIR Phase I: A Semantic Data-Driven Human Capital Recommendation System, NSF, $180,000, (2014).
17. Travel Grant Recipient: Attending Software Engineering Educators' Symposium. NSF, $1,300, (2012).

**INTERNAL GRANTS**

1. PI. FireBugs: Finding and Repairing Software Bugs with Security Patterns. UCRCA, $6,000, (2019).
2. PI. Curriculum Development Grant. UCAT, $1,500, (2014).
3. PI. Developing Course redesign: CS Principles. NCAT, $3,000, (2012).
4. Co-PI. A Cybersecurity Educational Platform by Automatically Generating Personalized Feedback, Nebraska Collaboration Initiative, NU Foundation, $143,773, (2020-2022).
5. Co-PI. Assessment Minigrant: General Education Assessment for Computer Science Principles, UNO Center for Faculty Excellence, $3,500, (2017-2018).
6. Co-PI. Assessment Minigrant: Program Assessment for the BS Computer Science Degree, UNO Center for Faculty Excellence, $3,500, (2016-2017).
7. Co-PI. Approaches for Improving the Effectiveness of the UNO Teaching and Learning Coding (TLC) Laboratory – Phase 2. Kelly Fund, $10,000, (2015-2016).
8. Co-PI. Developing an Online Master of Science Degree in Computer Science Education (MS-CSE). University of Nebraska Online Worldwide, $35,000, (2014-2015).
9. Co-PI. Peer-Led Team Learning to Improve the Effectiveness of the UNO Teaching and Learning Coding (TLC) Laboratory. UNO Tech Fee, $23,000, (2014-2015).
10. Co-PI. Approaches for Improving the Effectiveness of the UNO Teaching and Learning Coding (TLC) Laboratory. Kelly Fund, $10,679, (2014-2015).
11. Travel Grant Recipient: Attending Conference on Software Engineering Education and Training. UCAT, $800, (2013).

**PROGRAM COMMITTEES**

1. International Conference on Software and Data Technologies (ICSOFT) 2007-2012, 2018-2020.
2. International Conference on Software Engineering and Applications (ICSOFT-EA) 2013-2016.
3. International Conference on Software Paradigm Trends (ICSOFT-PT) 2013-2016.
4. International Conference on Data Technologies and Applications (DATA) 2012-2016.
5. Advanced Software Engineering and its Applications (ASEA) 2009, 2010, 2011.
6. Empirical Software Engineering and Measurement (ESEM) 2007, 2008, 2010, 2011.
7. International Conference on Progress in Informatics and Computing (PIC) 2010.
8. AOSD Early Aspects Workshop (EA) 2009.
9. APSEC Workshop on Accountability and Traceability in Global Software Engineering (ATGSE) 2007, 2008.
10. International Software Metrics Symposium (METRICS) 2004, 2005.

11. IEEE Workshop on Evolution of Large Scale Industrial Applications (ELISA) 2003.
12. International Conference on Software: Theory and Practice (ICS) 2000.

## CONFERENCE ORGANIZATION

1. Program Co-Chair, 21st Annual ACM Conference on IT Education (SIGITE 2020).
2. Track Chair, Engineering and Education Track, IEEE International Conference on Electro Information Technology (EIT 2017).
3. Local Arrangements Chair, ACM International Conference on Computing Education Research (ICER 2015).

## EDITORIAL BOARD

1. International Journal of Software Engineering and its Applications (IJSEIA) 2009-
2. International Scholarly Research Notices (ISRN): Software Engineering 2011-2013.

## JOURNAL REFEREE

ACM Transactions on Computing Education, IEEE Transactions on Software Engineering, ACM Transactions on Software Engineering and Methodology, Journal of Systems and Software, Information and Software Technology, Software Quality Journal, Empirical Software Engineering Journal, Sofware Practice and Experience, Automated Software Engineering Journal, ACM Transactions on Design Automation of Electronic Systems, SpringerPlus