Personal Statement

Chao Wang

SUMMARY

My long-term vision is to establish a world-class research program in my subarea of software engineering/formal methods. During the past five years, I worked toward this goal by focusing on three aspects: (1) creating a thriving research group, (2) building a synergistic education program, and (3) taking a balanced approach to service.

In terms of research, I attracted $4.6M competitive grants (personal credit: $2.9M), including an NSF CAREER award (2012) and an ONR Young Investigator Award (2013). I created a productive research team, consisting of five Ph.D. students and two Masters students currently. I also graduated two Ph.D. students and three Masters students. Our work has been published in many top venues and led to several best paper awards.

In terms of teaching, I taught "Introduction to Data Structures and Algorithms," a core undergraduate course in the computer engineering program (Virginia Tech) three times. I also developed and taught a new graduate course once, and taught two other existing graduate courses for a total of six times. For all these courses, my student teaching evaluation has been consistently and significantly above the average of all engineering faculty in Virginia Tech.

In terms of service, I served on the program committees of many flagship venues such as FSE, ICSE, ISSTA, and OOPSLA. I also served on seven NSF panels (including a virtual CRII early-career panel), one DOE career panel, and as an international reviewer of the Canadian NSERC Discovery grant. Locally, I served on many departmental committees, including three faculty search committees, one curriculum committee, and one graduate committee.

Overall, I have learned to stay focused on my long-term research vision while balancing the teaching and service tasks and making them mutually enhancing.

In the remainder of this document, I will report my activities in each individual task in detail.

RESEARCH STATEMENT

I focus on developing formal verification and program synthesis tools to improve the safety, reliability and security of critical computer systems (hardware, software, and embedded systems). Although my work often involves a wide range of formal analysis techniques such as computational logic, automaton theory, decision procedures, model checkers, and program analyzers, the focus has always been on the following two types of research problems:

- ***Verification***: Can we formally verify that a critical system is designed correctly, its implementations does not have runtime errors, and when applied in the real world, the system does not leak sensitive information through various side channels?
- ***Program synthesis***: Can we automatically generate software code from formal or informal specifications? What are the compelling applications for which synthesis offers more efficient, reliable, and secure solutions than manually written code?

For both types of problems, I am interested in developing practical solutions – methods and software tools that can scale up to large, real systems. My work has led to eleven US patents, several commercial/academic software tools, and papers in many flagship venues of my field (verification and program synthesis) including

- ICSE, FSE, ASE, ISSTA, PLDI, POPL (software engineering)
- CAV, TACAS, FM, FMCAD, DAC, ICCAD (formal methods)

Research and Development in Industry. Prior to becoming a tenure-track faculty member, I worked as a Research Staff Member in NEC Laboratories America, Inc. in Princeton, NJ (an industry research lab) for seven years (2004-2011), and developed many in-house hardware and software verification tools. For example, I contributed to the development of *VeriSol*, a tool for verifying temporal safety properties of hardware designs specified in the C language. *VeriSol* was made available commercially by NEC System Technologies as a property checker in the *CyberWorkBench (CWB)* environment. I also contributed to *F-Soft*, a tool for detecting common programming errors such as buffer overflow, null pointer deference, and mismatched lock-unlock in software code. *F-Soft* was deployed within the NEC Corporation (a Fortune-500 company). I received the 2006 NEC Laboratories Technology Commercialization award for my contributions to *F-Soft*. My work at NEC Labs also led to eleven U.S. patents, with another six U.S. patent applications pending.

After coming back to academia in 2011, I also established research collaborations with colleagues at Fujitsu Labs of America, Inc. and IBM Research. In the future, I plan to continue engaging these industry partners in problem sharing, curriculum development, technology transfer, and providing student internship opportunities.

Concurrent Software Verification. One focus of my ongoing research is developing new methods and tools to help programmers write concurrent software correctly and more efficiently. This is important because multicore CPUs are now pervasive, spanning from embedded systems and smartphones, to commodity PCs, all the way to high-end servers and distributed systems. To unleash the computing power of these CPUs, programmers have to write concurrent software. However, this is a difficult task due to the inherent non-determinacy of concurrent software and the astronomically large number of thread interleavings. My work in the past few years has been focused primarily on making concurrent software analysis easier by leveraging symbolic reasoning techniques and stateless model checking. For example, I developed the first dynamic partial order reduction (DPOR) algorithm for multithreaded software running under weak memory models such as TSO/PSO (prior works of others only handle software running on processors with sequential consistent memory). I also received the ACM Distinguished Paper award (FSE 2010) for a paper on “staged concurrent program analysis,” which introduced a method for more efficiently conducting symbolic program analysis. My students and I have published in flagship venues such as ICSE, FSE, ASE, ISSTA, PLDI, and POPL.

In the future, my plan is to not only *verify* software code written by developers, but also automatically *synthesize* software code with correctness, security, and efficiency guarantees. Of particular interest are implementations of high-performance concurrent data structures; the vision is to allow developers to focus on the functional correctness, while relying on the synthesis tool to handle implementation details, e.g., by automatically adapting high-level designs to various low-level computing platforms.

CPS and IoT Software Security. Another focus of my ongoing research is developing new methods and tools to improve the performance, reliability and security of software integrated into cyber-physical systems (CPS) and the Internet of things (IoT). In this broad class of emerging software applications, performance is often a major concern and implementation defects may lead to catastrophe. In this context, my students and I proposed a formal program synthesis based method for improving the precision of fixed-point arithmetic computations in resource-constrained controller software; the work won the 2013 FMCAD Best Paper award. We also developed a formal verification tool, named *SC Snifer*, for automatically detecting information leakage of embedded software through various “side channels” such as variations in power dissipation, execution time, or electronic-magnetic radiation. These side-channel leaks allow adversaries to quickly deduce internal secrets of embedded computing system, such as passwords, cryptography keys, and private settings of cyber-physical equipment and IoT devices. In addition to the formal verification tool, we also developed the first program synthesis tool for mitigating side-channel attacks using “provably secure” countermeasures. Our work has been published in top venues on formal methods such as CAV, TACAS, DAC, IEEE T-CAD, and ACM TOSEM.

In the future, my plan is to develop a more comprehensive formal verification and synthesis based solution for optimizing the performance as well as enhancing the security of CPS and IoT software. For mitigating side-channel attacks, in particular, I have been collaborating with domain experts (e.g., Professor Patrick Schaumont of Virginia Tech) to develop tools for detecting information leaks through the power, timing, and cache side channels; verifying side-channel resistant software using formal analysis; and refining the side-channel leakage models and evaluating them through hardware prototyping.

Safety of Hardware/Reactive Systems. I started my research career by working on formal verification of very large-scale integrated circuits (VLSI). Given a finite-state model of the integrated circuit and a temporal logic property, model checking is an algorithmic method for deciding if the model satisfies the property under all input conditions. However, the main problem of model checking is scalability: Since the model size grows exponentially with respect to the number of components, even state-of-the-art model checkers cannot directly handle large designs. In my doctoral research, I addressed the problem by developing fully automated abstraction techniques to bridge the scalability gap. My work won the 2003-2004 ACM Outstanding Ph.D. Dissertation award in Electronic Design Automation. This award, established by ACM SIGDA, is given each year to a Ph.D. dissertation that "*makes the most substantial contributions to the theory and/or application in the field of electronic design automation*." More recently, I have been collaborating with colleagues (Professor Michael Hsiao of Virginia Tech and researchers from Northrop Grumman Aerospace Systems) to help DARPA develop semi-formal verification techniques to further speed up the hardware verification process.

Beyond hardware verification, I also worked on developing new methods for protecting reactive systems from safety violations – whenever a safety-critical property cannot be formally verified, we generate a runtime enforcer to ensure that the combined system (composition of the system and the enforcer) never violates the property, even if the original system occasionally violates the property due to design defects or environmental disturbance. This is a more scalable approach to ensuring safety than classic techniques such as *model checking* and *reactive synthesis*, because the enforcer has to consider only a handful of safety-critical properties as opposed to the complete system. In the future, I plan to continue working along this direction, to develop a unified *runtime enforcer* synthesis framework, to robustly handle not only "safety" properties but also "liveness" and "fairness" properties of critical hardware/reactive systems.

TEACHING STATEMENT

I enjoy teaching at all levels. Of particular interest are graduate-level courses on formal verification and program synthesis and undergraduate-level courses in software engineering and programming languages. In the past five years, I have taught four different courses for a total of ten times – a teaching load comparable to that of a pre-tenured engineering faculty at USC. Below are the four courses I taught:

- "Introduction to Data Structures and Algorithms" – a core undergraduate course; three (3) times
- "Testing and Verification of Digital Systems" – a graduate course; five (5) times
- "Advanced Verification Techniques for Software Systems" – a new graduate course; once (1)
- "Multiprocessor Programming – a graduate course; once (1)

For all these courses, my student teaching evaluations (4.90 out of 6.0 for the undergraduate classes and 5.53 for the graduate classes) have been significantly above the averages of all Virginia Tech engineering faculty (4.68 for the undergraduate classes and 5.12 for the graduate classes). I also received two curriculum development grants.

Teaching Philosophy. My classroom teaching style is heavily influenced by my industry work experience. I believe in *Learning by Doing*. I have observed that, especially for introductory courses, hands-on project-based teaching is far more effective than traditional lecture-driven teaching. When I taught the sophomore-level programming course (Introduction to Data Structures and Algorithms), I designed five programming-intensive homework assignments and made them the focal point of my teaching. Feedback from students in three different semesters confirmed that it was significantly more effective than traditional lecture-driven teaching.

Synergy between Teaching and Research. I integrate research into my classroom teaching whenever possible. At the graduate level, this means creating new course materials and projects related to my ongoing research. At the undergraduate level, this means introducing, from time to time, exciting new development in our field to students. For example, due to the pervasive use of multicore CPUs, all undergraduate students in the computing field ought to understand the basic concepts of concurrency and parallel programing. However, curriculum development in higher education is always a slow process. Recognizing this need, I developed a reusable course module by leveraging my expertise in parallel programming and integrated it into the sophomore-level course. For this work, I received an Early Adopter Award from the NSF/IEEE-TCPP Curriculum Committee for Parallel Programming in 2014.

Undergraduate Mentoring. I enjoy working with undergraduate students and supervising their research projects. At Virginia Tech, I supported seven undergraduate students through my NSF REU supplement grants. Some students continued after the REU projects and subsequently did independent studies with me. Although I involve undergraduate students primarily for education purposes, some of these REU projects have been very successful. For example, we published three papers in reputable conferences (ASE 2013, ASE 2014 and VMCAI 2014) for which undergraduate students were the first- or second-authors. Our success was reported in an article on REU activities in the Virginia Tech ECE Department's 2014 annual report. One REU student who continued with me for his graduate study won the prestigious NSF Graduate Research Fellowship in 2016.

Diversity and Outreach. I am strongly sensitive to the issue of minorities and gender imbalance in engineering and have been supporting/mentoring female MS and PhD students. I also collaborated with the well-established Virginia Tech's Center for Enhancing Engineering Diversity (CEED) to give summer lectures to incoming freshman, as part of the "engineering student retention" program, and lectured in the first-year engineering student research seminar, ENGR 1014 (Fall'12). One student wrote to me afterward, saying "*I attended your presentation at the Engineering 1014 Seminar yesterday. I have to say it was by far the most interesting presentation, and the only one that has really pertained to me so far. The main reason to why I am writing (to) you, is that I would like to know if there is any way that I can become involved with you or one of your fields...*"

Beyond the department and the university, I also participated in many industry outreach activities through the VT Center for Embedded Systems for Critical Applications (CESCA), including giving lectures in the 2013 CESCA Security Symposium in Arlington, VA, which attracted many IT professionals in the Washington D.C. and National Capital region.

Overall, my teaching activities in the past five years aimed at bringing the best out of students so they can reach their full potentials. I plan to continue with these activities, and seek new and more effective pedagogy to promote both in-class and out-of-class learning.

SERVICE STATEMENT

I have been actively participating and leading various service efforts at the departmental, national, and international levels. I enjoy making contributions to my community.

Within the university, I have served on many committees, including three Faculty Search Committees at Virginia Tech (two in Computer Science and one in Electrical and Computer Engineering). I also served on the ECE Department's Curriculum Committee, Graduate Committee, and was elected to serve as a member of the Faculty Advisory Committee to Department Head. I was the Graduate Admission Officer of the Computer Engineering program (the largest in the nation according to ASEE 2010) for a year, and the Area Recruitment Representative for Graduate Admission for several years.

Outside the university, I have served on the technical program committees of numerous conferences and workshops (27 program committees in the past five years), including many top venues such as ICSE, FSE, OOPSLA, FM, and FMCAD. I also served on seven NSF panels (including a virtual panel on CRII mini-career proposals), one DOE CAREER proposal panel, and once as an international reviewer of the Canadian NSERC Discovery grant.

As a recognized researcher in my field, I was invited to give lectures in the 2014 International SAT/SMT Summer School (Europe). The focus of the summer school was on the emerging techniques for Satisfiability Modulo Theories (SMT) solvers, which have become the backbone of numerous applications such as automated verification, artificial intelligence, program synthesis, security, and product configuration. The summer school covered both foundational and practical aspects of SAT and SMT technologies and their applications. My lectures were based on the recent work of my research group on leveraging SAT/SMT solvers to formally verify "Concurrent Software and Cryptographic Software."

I also contributed to the research community by organizing and giving tutorials at conferences. For example, I co-organized a full-day tutorial for the 2015 International Conference on Formal Methods in Computer Aided Design. I gave a tutorial presentation on “security analysis of binary code” at the 2013 International Conference on Runtime Verification, and a tutorial presentation on “predictive analysis of concurrent software” at the 2012 International Conference on Runtime Verification.

Prior to becoming a tenure-track faculty member in 2011, I was also involved in organizing tutorials at several technical conferences, including a full-day tutorial on “embedded software verification” for the 2008 International Conference on Computer Aided Design.

Details of my services are highlighted in my CV.