Biography for Adjunct Lecture Award of the Year 2000

Ming-Hwa Wang, Ph.D.
Department of Computer Engineering
Santa Clara University

Ming-Hwa Wang received his B.Ed. degree in Industrial Education from National Taiwan Normal University, a special university which trains young college students to become high school teachers. That is the reason why Ming-Hwa enjoys teaching so much. He started to teach at Taipei Municipal Senior Industrial Vocational School, Taipei, Taiwan.

Ming-Hwa came to United States, and graduated from Rochester Institute of Technology with a M.S. degree in Printing Technology. After graduation, he went back to Taiwan and worked at China Engraving and Printing Works, the Mint of the Central Bank of Taiwan. He also served as an adjunct lecturer at NTNU, the university which he graduated from.

Having not taken any computer courses previously, Ming-Hwa started to teach computer classes from very basic courses to intermediate/advanced ones. All his computer knowledge and programming skills were self-taught through reading and teaching; therefore, he knows how to make students understand difficult subjects easily. He also believes that teaching is the best way to learn.

Ming-Hwa came back to United States again in September 1987, took six courses during the first semester at Illinois Institute of Technology, and passed the Ph.D. qualifying examination in November 1987. He earned his Teaching Assistantship in January 1988 and received his Ph.D. degree in Computer Science in May 1991.

During Ming-Hwa's Ph.D. study, he majored in distributed languages (compilers)/systems, formal method and parallel processing. The title of his Ph.D. Dissertation is "The Asynchronous MDC, a Relatively Low Level Distributed Computational Model for the Synchronous CSP, and its Denotational Semantics". The research involved:

- Introduced the formal definition of MDC (Message Driven Computing), which is a new Actors-like model (i.e., based on distributed data and pattern-driven control) with better pattern matching capabilities, by presenting its denotational semantics. New techniques are invented to formalize distributed objects, pattern matching, object-oriented features, type checking and asynchronous communications.
- Showed that the asynchronous MDC is a relatively low level model for the synchronous CSP (Communicating Sequential Processes):
  1. Theoretically. Simulated all CSP-like constructs (synchronous input/output, guarded command, alternative and repetitive constructs, interprocess communication protocol, remote procedure call, rendezvous, etc.) by MDC. Problems of distributed termination detection and busy waiting are also solved during simulation.
  2. Practically. Translated Joyce (a CSP-based language) into MDC/C (an MDC-based language), and proved that the translations are correct by getting the same results. The translation is done by dividing processes with synchronous message passing into light-weighted tasks, which only contain asynchronous message passing.
- Proved that all simulations are correct by using analysis techniques (reachability trees) of Petri nets.

Working in Silicon Valley as a computer engineer/manager after his Ph.D. (at Zycad, Silicon Graphics, NeoParadigm Labs, Dahan Information Systems, Raycer Graphics, Procket Networks, Quicksilver Technology, Texas Instruments, Micron Technology, Advanced Micro Circuit Corp, Conexant, Cisco, and now Sizmek/RocketFuel), Ming-Hwa also joined Santa Clara University as an adjunct lecturer in Fall 1996. He teaches Computer Graphics, Distributed Computing, Data Structures and Computer Algorithms, Formal Languages and Compiler Constructions, Code Generation and Optimization, Computer Networks, Internet Architecture and Protocols, Interactive Multimedia and Game Programming, SoC Verification, Distributed Systems, Big Data, Cloud Computing, Data Mining, Information Retrieval, Machine Learning, Semantic Web, Natural Language Processing, and Artificial Intelligence.

Ming-Hwa puts all class notes on his web page (which is regularly updated), keeps programming assignments and examination questions different for every quarter, always responds to students' questions promptly through phone or email, and knows how to motivate students to learn and to work hard. In addition, he developed a fully automatic software system to automate programming assignment submission, testing, grading, and score reporting. His evaluations from students average above 4.0 on the 5.0 scale. He received the Adjunct Lecturer of the Year Award in 2000.