YOONSIK CHEON

The University of Texas at El Paso
Department of Computer Science
500 West University Avenue
El Paso, TX 79968-1518
Phone: +1-915-747-8028
FAX: +1-915-747-5030
E-mail: ycheon@utep.edu
URL: http://www.cs.utep.edu/~cheon

RESEARCH INTERESTS

Formal methods: specification languages, specification and verification techniques, and tool support

Programming: programming languages, formal semantics, and programming methodology

EDUCATION

Ph.D., Computer Science, Iowa State University, Ames, Iowa, May, 2003

M.S., Computer Science, Iowa State University, Ames, Iowa, 1991

B.S., Computer Science, Korea University, Seoul, Republic of Korea, 1989

EXPERIENCE

2003–Present: **Assistant Professor, Computer Science, University of Texas at El Paso**, El Paso, Texas.

1995–2003: **Senior Researcher, Electronics and Telecommunications Research Institute (ETRI)**, Taejon, Republic of Korea.

1995–1996: **Researcher, ETRI**, Taejon, Republic of Korea.

TEACHING

University of Texas at El Paso, El Paso, Texas — Fall 2003 – Present

CS 5383-Topics in Software Assurance
CS 5382-Topics in Software Development
CS 5381-Topics in Software Design
CS 4311-Software Engineering: Design and Implementation
CS 4310-Software Engineering: Requirements Engineering
CS 3360-Design and Implementation of Programming Languages
CS 3331-Advanced Object-Oriented Programming

RECENT PUBLICATIONS

Myoung Yee Kim and Yoonsik Cheon. A Fitness Function to Find Feasible Sequences of Method Calls for Evolutionary Testing of Object-Oriented Programs. To appear in *International Conference on Software Testing, Verification, and Validation, Norway, April 9-11, 2008.*

Yoonsik Cheon. Abstraction in Assertion-Based Test Oracles. In *Proceedings of the Seventh International Conference on Quality Software, Portland, Oregon, USA, October 11-12, 2007,* pages 410-414, IEEE Computer Society.

Yoonsik Cheon. Automated Random Testing to Detect Specification-Code Inconsistencies. In *Proceedings of the 2007 International Conference on Software Engineering Theory and Practice, July 9-12, 2007, Orlando, Florida, U.S.A.,* pages 112-119.

Yoonsik Cheon and Carlos E. Rubio-Medrano. Random Test Data Generation for Java Classes Annotated with JML Specifications. In *Proceedings of the 2007 International Conference on Software Engineering Research and Practice, Volume II, June 25–28, 2007, Las Vegas, Nevada,* pages 385-392.

Hyotaeg Jung, Carlos E. Rubio-Medrano, Eric Wong, and Yoonsik Cheon. Architectural Assertions: Checking Architectural Constraints at Run-Time. *The 6th International Workshop on System and Software Architectures*, pages 604-607. Published in *Proceedings of SERP 2007, Volume II, June 25-28, Las Vegas, Nevada.*

Poonam Agarwal, Carlos E. Rubio-Medrano, Yoonsik Cheon, and Patricia J. Teller. A Formal Specification in JML of the Java Security Package. In Khaled Elleithy (ed.), *Advances and Innovations in Systems, Computing Science, and Software Engineering*, pages 363-368, Springer, 2007.

Yoonsik Cheon and Ashaveena Perumandla. Specifying and Checking Method Call Sequences of Java Programs. *Software Quality Journal*, 15(1):7-25, March 2007.

Angelica B. Perez, Yoonsik Cheon, and Ann Q. Gates. Canica: An IDE for the Java Modeling Language. In *Proceedings of International Conference on Software Engineering and Applications, November 13-15, 2006, Dallas, TX*, pages 32-37.

James C. Browne, Calvin Lin, Kevin Kane, Yoonsik Cheon, and Patricia J. Teller. In *20th International Parallel and Distributed Processing Symposium (IPDPS 2006), Proceedings, 25-29 April 2006, Rhodes Island, Greece*, IEEE 2006.

Yoonsik Cheon and Myoung Kim. A Fitness Function for Modular Evolutionary Testing of Object-Oriented Programs. In *Genetic and Evolutionary Computation Conference, Seattle, WA, USA, July 8-12, 2006*, pages 1952-1954, ACM Press, 2006.

Yoonsik Cheon and Gary T. Leavens. A Contextual Interpretation of Undefinedness for Runtime Assertion Checking. In *AADEBUG 2005, Proceedings of the Sixth International Symposium on Automated and Analysis-Driven Debugging, Monterey, California, September 19–21, 2005*, pages 149-157. ACM Press, September 2005

Lilian Burdy, Yoonsik Cheon, David R. Cok, Michael Ernst, Joe Kiniry, Gary T. Leavens, K. Rustan M. Leino, and Erik Poll. An Overview of JML Tools and Applications. *International Journal on Software Tools for Technology Transfer*, 7(3):212-232, June, 2005.

Yoonsik Cheon and Ashaveena Perumandla. Specifying and Checking Method Call Sequences in JML. *Proceedings of the 2005 International Conference on Software Engineering Research and Practice*, Las Vegas, Nevada, June 27-29, 2005, pages 511-516, 2005.

Yoonsik Cheon, Myoung Yee Kim, and Ashaveena Perumandla. A Complete Automation of Unit Testing for Java Programs. *Proceedings of the 2005 International Conference on Software Engineering Research and Practice*, Las Vegas, Nevada, June 27-29, 2005, pages 290-295, 2005.

Yoonsik Cheon, Gary T. Leavens, Murali Sitaraman, and Stephen Edwards. Model Variables: Cleanly Supporting Abstraction in Design By Contract. *Software—Practice and Experience*, 35(6):583–599, May, 2005.

Gary T. Leavens, Yoonsik Cheon, Curtis Clifton, Clyde Ruby, and David R. Cok. How the Design of JML Accommodates Both Runtime Assertion Checking and Formal Verification. *Science of Computer Programming*, 55(1-3):185-208, March 2005.

Yoonsik Cheon, Yoshiki Hayashi, and Gary T. Leavens. A Thought on Specification Reflection. In N. Callaos, W. Lesso, and B. Sanchez, *The 8th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2004), July 18-21, 2004, Orlando, Florida, USA*, *Volume II, Computing Techniques*, pages 485–490, July 2004.

Gary T. Leavens, Yoonsik Cheon, Curtis Clifton, Clyde Ruby, and David R. Cok. How the Design of JML Accommodates Both Runtime Assertion Checking and Formal Verification. In *First International Symposium on Formal Methods for Components and Objects (FMCO 02), Leiden, The Netherlands, November 5-8, 2002*, page 262-284. Volume 2852 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 2003.

Lilian Burdy, Yoonsik Cheon, David R. Cok, Michael Ernst, Joe Kiniry, Gary T. Leavens, K. Rustan M. Leino, and Erik Poll. An Overview of JML Tools and Applications. In Thomas Arts and Wan Fokkink (editors), *Eighth International Workshop on Formal Methods for Industrial Critical Systems (FMICS '03)*, pp. 73-89. Volume 80 of Electronic Notes in Theoretical Computer Science, Elsevier, June, 2003.

Yoonsik Cheon and Gary T. Leavens. A Simple and Practical Approach to Unit Testing: The JML and JUnit Way. In Boris Magnusson (ed.), *ECOOP 2002 – Object-Oriented Programming, 16th European Conference, Malaga, Spain, June 2002, Proceedings.* Volume 2374 of *Lecture Notes in Computer Science*, pages 231-255. Springer-Verlag, 2002.

Yoonsik Cheon and Gary T. Leavens. A Runtime Assertion Checker for the Java Modeling Language (JML). In Hamid R. Arabnia and Youngsong Mun (eds.), *International Conference on Software Engineering Research and Practice (SERP '02), Las Vegas, Nevada*, pages 322-328. CSREA Press, June 2002.

GRANTS

PI, Improving Dependability of Software through Rigorous Testing against Requirements, *Department of Defense*, $125,000, 12/01/2007–09/30/2008.

PI, Collaborative Research: A JML Community Infrastructure—Revitalizing Tools and Documentation to Aid Formal Methods Research, *National Science Foundation*, CNS-0707874, $100,000, 07/15/2007–07/14/2010.

Co-PI, Training and Software Support for the Senior Design Course in Software Engineering at The University of Texas at El Paso, *Rockwell Collins*, $20,000, June 2007.

PI, Collaborative Research: Unification of Verification and Validation Methods for Software Systems, *National Science Foundation*, CNS-0509299, $167,999, 09/01/2005–08/31/2008.

PI, Automated Testing of Object-Oriented Programs, *UTEP University Research Institute*, $5,100, 09/2004-8/2005.

PROFESSIONAL MEMBERSHIPS

Association for Computing Machinery (ACM)

IEEE Computer Society

PROFESSIONAL ACTIVITIES

Program committee for *International Conference on Software Engineering and Knowledge Engineering* (2008), *International Workshop on Software Test Evaluation* (2008, 2007), *International Conference on Formal Engineering Methods* (2008, 2007, 2006), *International Conference on Software Engineering Advances* (2008, 2007, 2006), *International Conference on Computer and Information Science* (2008), *International Workshop on Software Architecture Research and Practice* (2008, 2007), *International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing* (2008, 2007), *International Conference on Programming Languages and Compilers* (2006), *International Conference on Software Engineering Research and Practice* (2006), *ACM Symposium on Applied Computing* (2006), *Workshop on Formal Techniques for Java-Like Programs* (2006), and *Information Technology Symposium of US-Korea Conference* (2006).

Journal referee for *Science of Computer Programming* (2006, 2008), *IEEE Transactions on Software Engineering* (2003, 2005), *Journal of Automated Software Engineering* (2003), *Software Practice and Experience* (2003), *Journal of Korean Information Science Society* (1995–1997), *Journal of Object Technology* (2006, 2007), *Journal of Systems and Software* (1992), *Journal of Supercomputing* (2006), and *Reliable Computing* (2005).

Conference referee for *SNPD* (2007), *ACM SAC* (2006), *ECOOP* (2004, 2003), *FSE* (2004), *FTfJP* (2006), *ICSEA* (2006), *IEEE DRAT* (2005), *RV* (2005), and *OOPSLA* (2006, 2004, 1993).

SOFTWARE ARTIFACTS

JET is a unit testing tool for Java, automating each step of testing from test data generation to test execution and result determination. It is available from `http://www.cs.utep.edu/~cheon/download`.

UTJML is an extension to JML (see below) that provides a research platform for interface specifications, runtime assertion checking, program testing, and verification and validation. The UTJML tool is available from `http://www.cs.utep.edu/~cheon/download`.

JML is a formal behavioral interface specification language for Java. JML documentation and tool suite are available from `http://www.jmlspecs.org`.

Larch/C++ is a Larch-style formal behavioral interface specification language for C++. More information on Larch/C++ is available from `http://www.eecs.ucf.edu/~leavens/larchc++.html`.

Larch/Smalltalk is a Larch-style formal behavioral interface specification language for Smalltalk. More information is available from `http://www.eecs.ucf.edu/~leavens/larchSmalltalk.html`.