

CURRICULUM VITA – YAPING JING

CONTACT INFORMATION Yaping Jing, Tel: (+1) 410-543-6470, Email: yxjing@salisbury.edu.
URL: <https://faculty.salisbury.edu/~yxjing/>

EDUCATION Ph.D. Computer Science, Iowa State University (ISU), Ames, Iowa. Dec. 2015.
Advisor: [Prof. Andrew Miner](#).

M.S., B.S., Computer Science, ISU, May 2005 and May 2002, respectively.

CURRENT **Associate Professor, Computer Science, Salisbury University, Salisbury, MD.
(08/2020-Present)**

**Assistant Professor, Mathematics and Computer Science, Salisbury University,
Salisbury, MD. (08/2016-08/2020)**

- COSC386 – Database Systems. Fall 2016-Present, Every Semester, 3 Sections Per Year.
- COSC120 – Introduction to C++. Spring 2018, Spring 2019
- COSC119 – Web Development. Spring 2017, Spring 2018, Spring 2019
- COSC117 – Programming Fundamentals. Fall 2016, Fall 2017, Fall 2018
- COSC385 – Directed Study for Web Development (Keith Livingston) Summer 2019
- COSC390 – Undergraduate Research Projects and Student Participants
 - *Automated Grading System*, by Ian Thomas (Fall 2018, Spring 2019), Ian Schwartz (Spring-Fall 2017, Spring 2019), Lela Bones (Spring 2017)
 - *Neural Networks in Gaming*, by Keith Flagg, Justin Ventura, and Ian Thomas (Fall 2019-Spring 2020).
 - *A Case Study of CTML on Mobile Manipulator*, by Omar Aboul-Enein (Fall 2017, Spring 2018)
 - *A Web Based Voting Tool*, by Garret Spicer-Davis (Spring 2017)

RESEARCH AREA

- Stochastic Modeling and Analysis on Complex Systems.
- Probabilistic Model Checking and Performance Evaluation.
- Database Systems
- Tool Development

PUBLICATIONS Omar Aboul-Enein, Yaping Jing, and Roger Bostelman (2020). “*Formalizing Performance Evaluation OF Mobile Manipulator Robots Using CTML.*” Proceedings of the ASME 2020 International Mechanical Engineering Congress and Exposition (IMECE). Accepted.

Yaping Jing and Andrew S. Miner (2018). “*Computation Tree Measurement Language (CT-ML).*” Formal Aspects of Computing (FAC 2018). 30(3-4), 443-462. Print ISSN: 0934-5043. Online ISSN: 1433-299X. Springer, London. <https://doi.org/10.1007/s00165-018-0457-3>.

Yaping Jing and Andrew S. Miner (2018). “*Action and State Based Computation Tree Measurement Language and Algorithms.*” In: A. Horvath & A. McIver (Eds.), 15th International Conference on Quantitative Evaluation of SysTems (QEST 2018), LNCS 11024, pp. 190-206. Springer Cham. https://doi.org/10.1007/978-3-319-99154-2_12.

Joseph Anderson, Jathan Austin, Yaping Jing, Lisa Schneider, Ryan Shifler, and Sarah Wesolowski (2018). “*Faculty Writing Groups for Mathematicians.*” Mathematical Association of America (MAA) Focus, October/November, pp. 36-36. <https://www.maa.org/press/periodicals/maa-focus>.

Yaping Jing (2015). “A formal language towards the unification of model checking and performance evaluation”. Ph.D. Dissertation. ISU 2015. Paper 14855.

Andrew Miner and Yaping Jing (2010). “A Formal Language towards the Unifications of Model Checking and Performance Evaluation.” In: K. Al-Begain, D. Fiems, and W. Knottenbelt (Eds.) 17th International Conference on Analytical & Stochastic Modelling Techniques & Applications (ASMTA 2010), LNCS 6148, pp. 130–144. Springer, Heidelberg. https://doi.org/10.1007/978-3-642-13568-2_10.

Yaping Jing and Alex Travesset (2006). “Crystalline Particle Packings on Spheres.” American Physical Society Abstracts, 2006.

Yaping Jing (2005). “Towards the Representation of Design Patterns as Design Components.” M.S. Thesis. Call No. ISU 2005. J564.

PAPER REVIEW External Reviewer for the National Institute of Standards and Technology (NIST) Intelligent Systems Division on the journal paper titled “*Measurement of Advanced Mobile Manipulator Performance Towards Assembly Applications.*” (Jan. 2018) By Roger Bostelman, Ya-Shian Li-Babound, SooCheol Yoon, Mili Shah, Omar Aboul-Enein.

CONFERENCE PRESENTATION **International Conference on the Quantitative Evaluation of Systems**, September 4-7, 2018. Beijing, China. Presentation Title: “*Action and State Based Computation Tree Measurement Language and Algorithms.*” <http://www.qest.org/qest2018/index.html>.

International Conference on the Analytical & Stochastic Modeling Techniques and Applications, June 14-16, 2010. Cardiff, UK. Presentation Title: “*A formal language towards the unification of model checking and performance evaluation.*” <https://at-web1.comp.glam.ac.uk/asmta2010/>.

PROJECT DIRECTOR Omar Aboul-Enein & Yaping Jing. “*A Case Study of CTML towards Performance Evaluation of Mobile Manipulators.*” (2018) A **departmental honor project**.

SU STUDENT RESEARCH CONFERENCE 2019 Salisbury University Student Research Conference (SUSRC) faculty advisor for the project titled: “*Using Asynchronous technology to produce automated feedback for software-based assignments*”, presented by Ian Thomas and Ian Schwartz.

GRANT SU Foundation: Awarded \$1000 (max) for the conference traveling to QEST 2018.

PAST WORK EXPERIENCE **Visiting Faculty, Department of Math and Computer Science, Whitman College. Walla Walla, Washington. 08/2015-07/2016.**

- CS167 – Introduction to Python. Fall 2015
- CS270 – Data Structures Using C++. Fall 2015, Spring 2016
- CS400 (newly developed) – Design and Analysis of Algorithms. Spring 2016

Visiting Faculty, Department of Computer Science, Illinois Wesleyan University. Bloomington, IL. 01/2014-07/2015.

- CS314 – Database Systems. Spring 2014
- CS135 – Discrete Mathematics and Functional Language. Spring 2014
- CS128 – Data Structure and Algorithms in C++. Spring 2014, Fall 2014, Spring 2015
- CS355 – Programming Languages. Fall 2014
- CS126 – JavaScript Programming. Fall 2014

- CS127 – Introduction to Python. Spring 2015
- CS370 (newly developed) – Fundamentals of Software Engineering. Spring 2015

Research Assistant, Mechanical Engineering Department, ISU. Summer 2011.

Research Assistant, Physics Department, ISU. 05/2005-06/2006.

Software Engineer, Division of Engineering, Mayo Foundation, Rochester, MN. Summer 2006 and Summer 2007, respectively.

Teaching Assistant, Computer Science Department, Iowa State University

CS511 – Design and Analysis of Algorithms (network flow, etc.). Fall 2011.

CS362 – Object-Oriented Analysis and Design. Spring 2011, Fall 2004.

CS309 – Software Development Practices (software life cycle, agile). Spring 2010.

CS461/561 – Principles of Relational Databases (JDBC, Hibernate). Fall 2009.

CS252 – Linux Operating System Essentials (Ubuntu, Fedora). Fall 2009, Fall 2010.

CS440/540 – Principles of Compilers. Spring 2009.

CS409/509 – Requirements Engineering. Fall 2008.

CS103 – Computer Literature and Practice. Fall 2006, Spring 2007.

CS227 – Introduction to Object Oriented Programming (C++). Spring 2005.

CS342 – Principles of Programming Languages (Scheme). Spring 2004.

CS228 – Introduction to Data Structures (C++). Fall 2003.

SOFTWARE
TOOLS

Performance-Dependability Analyzer (Implemented in Java), 03/2009-03/2012.

Given a computation tree measurement language (CTML) specification that encodes path-based performance-dependability (reliability, survivability, availability, etc.) related queries, and a structure of Markov chain model, the tool outputs quantitative real values in the range of $[0, \infty)$, rather than merely 0 or 1, for the set of queries automatically. Currently, the tool has full support of CTML specifications, which covers probabilistic computation tree logic (PCTL) and a subset of probabilistic linear time logic (PLTL) that can not be expressed by PCTL (with polynomial time and space in the size of the model). This software is developed and tested on graphs with millions of states and edges.

SAT Solver (Implemented in Java). 07/2010-12/2012.

Given a SAT formula of n variables and m conjunctive clauses, where each clause is a disjunction of a subset of n variables or its negation (a.k.a. literals), the tool prints out a sequence of variables or its negation if there exists an assignment to the variables such that the whole SAT formula is evaluated to *true*; otherwise it reports No.

Markov Chain Solver (in Java). Spring 2007.

A standard Markov chains solver for transient analysis (the probability distribution when time is bounded) and limiting behavior analysis (the probability distribution when times goes to infinity), where the underlying Markov chains are discrete time Markov chains (DTMCs) and continuous time Markov chains (CTMCs). This software also supports discrete and continuous time stochastic logics (PCTL, CSL).

MISC.
SOFTWARE

General Stochastic Petri Net (GSPN) Simulation Library (in C/C++), Fall 2007.

GSPN simulation is an alternative of performance modeling for the analysis of complex systems. The advantage of this approach is that it doesn't incur state explosion problem, albeit it does not produce as exact answers as performance modeling approach does. In the context of discrete-event simulation, we can find one to one mapping between GSPN and its underlying stochastic process. The purpose of this project is to develop a discrete-event simulation library that collects statistics like confidence intervals, batch means, throughput, etc. for GSPN models.

Design Patterns (Implemented in C++), Fall 2003-Fall 2004

Promoted a subset of GoF's 23 design patterns (such as chain of responsibility, singleton, observer, etc.) into design components for code reusability.

Cellular Automata Compiler (in Java), Spring 2004.

Cellular automata (CA) were originally conceived by Stanislaw M. Ulam and John von Neumann in the 1940s to provide a formal framework for investigating the behavior of complex, adaptive systems in which time and space are discrete. A CA space consists of a grid of cells, each can be in one of N finite states at any time. A cell can be in one of two states: 0 (dead) or 1 (live), and its next state is determined by the current state of itself and its neighborhood. The focus of this project is to make a compiler including a *lexer*, a *parser*, and *code generator*, by using any high level programming languages (Java or C/C++) and the corresponding tools (Jlex/cup for Java, or yacc for C/C++). Additionally, I developed a graphical user interface for testing the given set of CA rules more intuitively.

COMPUTER
SKILLS

Operating systems Mac OS X, UNIX, Linux, Windows, and Solaris.

Servers and databases Apache2, openssh, subversion, NFS, and MySQL.

Programming, scripting and markup languages Java, C/C++, Scheme, Fortran, Visual Basic, Haskell, SQL, MIPS assembly, PHP, Javascript, HTML, and Bash.

Libraries and Tools STL, OpenGL, Latex, Xcode, BLAS, CBLAS, Visual Studio, Eclipse.

MEMBERSHIP

Association for Computing Machinery (ACM) member (2016-present).

ACM is the world's largest educational and scientific computing society; it delivers resources that advance computing as a science and a profession. <https://www.acm.org>.

Institute of Electrical and Electronics Engineers (IEEE) member (2016-present) and (2008-2012). IEEE is the world's largest technical professional organization for the advancement of technology. <https://www.ieee.org>.

Special Interest Group on Computer Science Education (SIGCSE) member (2014-2016). SIGCSE is a global forum for computer science educators to discuss research and practice related to the learning and teaching of computing. <https://sigcse.org/sigcse/>.

UNIVERSITY
SERVICE

Spring 2019, Dean of Henson School Search Committee, SU, Salisbury, MD.

Fall 2016-Present, Faculty Advisor for Upsilon Pi Epsilon SU Chapter, Salisbury, MD.

Spring 2020- Present, Computer Science Internship Coordinator, SU, Salisbury, MD.

Fall 2017-Present, Henson Research and Development Committee, SU, Salisbury, MD.

Fall 2017-Summer 2020, Cultural Diversity and Inclusion Committee, SU, Salisbury, MD (Main achievement: got the space of "The Center for Equity, Justice, and Inclusion" in the Backwell Hall Building).

Fall 2017-Summer 2020, Fiscal Advisory Committee, SU, Salisbury, MD.

Spring 2010, Administrative Support Programs Advisory Committee, ISU, Ames, IA.

COMMUNITY
SERVICE

Summer 2015, Science Camp for Youth, Bloomington, IL.