INDIANA UNIVERSITY

University Graduate School
2009-2010
Academic Bulletin

Computer Science

School of Informatics and Computing Bloomington

Dean
Robert Schnabel*

Associate Dean of Graduate Studies and Research
Geoffrey C. Fox*

Director of Graduate Studies
Amr Sabry*

Departmental E-mail
info@cs.indiana.edu

Departmental URL
www.cs.indiana.edu

Graduate Faculty

(An asterisk [*] denotes membership in the University Graduate School faculty with the endorsement to direct doctoral dissertations.)

College Professor of Cognitive Science and Computer Science Distinguished Professor of Computer Science
Douglas R. Hofstadter*

Professors
Randall Beer*, Randall Bramley*, Geoffrey Brown*, J. Michael Dunn* (Emeritus), R. Kent Dybvig*, Geoffrey C. Fox*, Daniel P. Friedman*, Stanley Hagstrom* (Emeritus), Andrew J. Hanson*, Douglas R. Hofstadter*, Steven D. Johnson*, David B. Leake*, Daniel M. Leivant*, Andrew Lumsdaine*, Michael A. McRobbie*, Franklin Prosser* (Emeritus), Paul W. Purdom*, Edward L. Robertson* (Emeritus), Amr A. Sabry*, Robert B. Schnabel*, George Springer* (Emeritus), Dirk Van Gucht*, David E. Winkel* (Emeritus), David S. Wise*

Associate Professors
Kay Connelly*, Michael Gasser*, Christopher T. Haynes*, Filippo Menczer*, Jonathan Mills*, Beth A. Plale*, Gregory J. E. Rawlins*, Matthias Scheutz*

Assistant Professors
Arun Chauhan, Minaxi Gupta*, Kris Hauser, Raquel Hill*, Yuqing Melanie Wu*

Adjunct Professors
David McCarty* (Philosophy), Robert Port* (Emeritus, Linguistics)

Adjunct Associate Professor
Larry Moss* (Mathematics)

Adjunct Assistant Professor
Chen Yu* (Psychology)

Degrees Offered
Master of Science and Doctor of Philosophy

Special Departmental Requirements
(See also general University Graduate School requirements.)

Admission Requirements
Admission to all graduate programs is by approval of the department's graduate admission committee. Requirements for admission: baccalaureate degree (not necessarily with a concentration in computer science) and Graduate Record Examination (subject test also desirable). Undergraduate course prerequisites may be satisfied by equivalent or more advanced courses, and in some cases by professional experience. Prerequisites common to all graduate requirements are course work in computer structures and organization, discrete structures and computing theory, and data structures.

Master of Science Degrees

(See School of Informatics and Computing Bulletin)

Doctor of Philosophy Degree

Course Requirements
A total of 90 credit hours of graduate-level course work is required. These courses are defined as any course listed in this bulletin that carries graduate credit. Note that no computer science courses in the A500-A598 range may be counted toward the 90 credit-hour requirement or toward the 24 credit-hour requirement specified as follows.

Computer Science Course Requirements
Ph.D. candidates must take at least 24 credit hours, normally eight courses, in computer science at the 500 level or above, subject to the following conditions:
1. P requirement: At least one must be a P course, with a

substantial programming or software development component.

2. Essentials requirements: Of the eight courses, there must be at least one course in Foundations/Logic (indicated by middle digits 0/1) and one course in Software/Hardware Systems (indicated by middle digits 3/4). Both of these courses must be passed with a minimum grade of B+.

3. Area distribution requirements: Of the eight courses, there must be at least one course each in six of the nine areas (indicated by the middle digit 0-8 in advanced computer science courses).

4. Research course conditions: The Y790 course is excluded from these six area courses and cannot fulfill the P requirement, but up to 6 hours of Y790 may be counted toward the 24 credit-hour requirement. Y890 and G901 are excluded from the 24 credit hours in this requirement.

5. A grade average of B (3.0) is required for computer science courses, in addition to the University Graduate School's requirement of a B (3.0) average for all courses taken.

Minor Area Requirement
Three options are available:

1. An external minor awarded by another Indiana University department or graduate program approved by the Computer Science Department.

2. An internal minor: 9 computer science credits, in courses other than reading and research, and in an area other than the student's specialization. The area and the courses must be approved by the student's advisory committee. These 9 credits cannot be counted toward the six-course requirement.

3. An individualized interdisciplinary minor, as prescribed by this bulletin: at least 12 credits spanning at least two departments, to be recommended by the student's advisory committee and approved by the dean in advance of any course work.

Qualifying Examination
The qualifying examination is given by the first semester of the student's third year in the program. This examination is administered by the advisory committee and is expected to have a written and an oral component.

Thesis Proposal
Given after completion of the qualifying examination (written and oral), consisting of an oral presentation of a written research plan for the dissertation. This examination is given by the research committee.

Dissertation
A written elaboration of significant original research, which must be successfully presented to the research committee in a defense of dissertation as described in this bulletin.

Ph.D. Minor in Computer Science

Doctoral students in other departments may complete a minor in computer science by satisfying one of the following options:

1. Three computer science courses totaling not fewer than 9 credit hours at the 500 level or above. A500-level courses and 400-level courses are excluded with these exceptions: A595 (B401), B403, P423, P436, P438, B441, P442, and B443 are approved for graduate credit toward the Ph.D. minor.
2. A593, A594, and any two courses totaling 6 credit hours or more from the list: A595, A596, plus the computer science courses meeting the requirements of the first option.

Courses

Most of the Computer Science department's courses at the 400 level and above are classified into four areas:

Foundations (middle digit 0 or 1)
Programming Languages (middle digit 2)
Systems (middle digit 3 or 4)
Applications (middle digit 5, 6, 7, or 8)

General courses not associated with a specific area are numbered with a middle digit 9. Courses that involve a major programming project are designated as "programming-in-the-large," and carry a course number with letter designation P. A500-level courses may be taken for graduate credit by graduate students in other departments, but are not applicable toward a graduate degree in computer science. Of these, only A593, A594, A595, and A596 are available for graduate credit toward the Ph.D. minor.

A504 Introductory C++ Programming (2 cr.) P: Programming experience. Topics include aspects of C++ that are not object-oriented, basic data structures, standard libraries, and Unix tools for project management. Credit not given for both A504 and either A304, A597, A592, C212, H212, or BUS K201.

A506 Object-Oriented Programming in C++ (2 cr.) P: Either A201, A304, A504, or A597. Topics include objects, classes, encapsulation, inheritance, polymorphism, templates, and exceptions. Credit not given for both A506 and either A306, A202, A592, A598, C212, or H212.

A521 Computing Tools for Scientific Research (3 cr.) C: Math 118 or higher required; Math M211 recommended. Introduction to computer-based tools useful for analysis and understanding of scientific data. Basic methods of computation, data processing, and display insystems such as Matlab combined with elementary practical C/C++ programming. Techniques to support customized scientific research tasks, with particular emphasis on biological, neural, and behavioral sciences. Lecture and laboratory.

A538 Network Technologies and Administration (3 cr.) P: A110, EDUC W200, or equivalent computer literacy. Introduction to network principles and current network technology, both hardware and software. Network administration tools and techniques. Laboratory provides practical experience. Credit not given for both A538 and A338.

A546 User-Interface Programming (3 cr.) P: Either A201, A202, A306, C212, A506, A597, A598, or equivalent experience. Learn

to prototype and build graphical user interfaces for computer applications. Contemporary software design methodology. Students design and implement prototype interfaces to applications provided by the instructor. Extensive use will be made of both commercial and experimental software tools. Lab fee. Credit not given for both A546 and A346.

A548 Mastering the World Wide Web (3 cr.) P: Two semesters of programming experience or equivalent, and some knowledge of operating systems. Project-oriented course leading to ability to maintain a Web site with full functionality. Topics include background on Internet network protocols and programming, Web server administration, advanced Web design and authoring, Web protocols, interfacing services into the Web. Lab fee. Credit not given for both A548 and A348.

A590 Topics in Programming (1-2 cr.) Eight-week courses designed to provide foundations for using modern programming tools for applications and web development. Lecture and lab. May be repeated for a maximum of six credits.

A591 Introduction to Computer Science (3 cr.) A first course in computer science for those intending to take advanced computer science courses. Introduction to programming and to algorithm design and analysis. Using the Scheme programming language, the course covers several programming paradigms. Lecture and laboratory. Credit not given for both A591 and C211.

A592 Introduction to Software Systems (3 cr.) P: Programming experience. Design of computer software systems and introduction to programming. Topics include the C++ programming language and its data structure facilities; building and maintaining large projects; shell tools, and system calls. Introduction to object-oriented programming. Lecture and laboratory. Credit not given for both A592 and C212.

A593 Computer Structures (3 cr.) P: A592. Structure and internal operation of computers. The architecture and assembly language programming of a specific computer are stressed, in addition to general principles of hardware organization and low-level software systems. Lecture and laboratory. Lab fee. Credit not given for both A593 and C335. May be applied toward the Ph.D. minor.

A594 Data Structures (3 cr.) P: A592. P or C: C241 and A593. Systematic study of data structures encountered in computing problems; structure and use of storage media; methods of representing structured data; and techniques for operating on data structures. Lecture and laboratory. Credit not given for both A594 and C343. May be applied toward the Ph.D. minor.

A595 Fundamentals of Computing Theory (3 cr.) P: C241. P or C: C212. Fundamentals of formal language theory, computation models and computability, the limits of computability and feasibility, and program verification. Credit not given for both A595 and B401. May be applied toward the Ph.D. minor, graduate credit available for CS M.S. candidates with special permission.

A596 Programming Languages (3 cr.) P: A594. Systematic

approach to programming languages. Relationships among languages, properties and features of languages, and the computer environment necessary to use languages. Lecture and laboratory. Credit not given for both A596 and C311. May be applied toward the Ph.D. minor.

A597 Introduction to Programming I (3 cr.) Fundamental programming constructs, including loops, arrays, classes, and files. General problem-solving techniques. Emphasis on modular programming, user-interface design, and developing good programming style. Credit not given for both A597 and A201.

A598 Introduction to Programming II (1.5-3 cr.) P: A597, A201, A504, or A304. Advanced programming techniques: user-defined functions and types, recursion vs iteration, parameter-passing mechanisms. Classic abstract data types and algorithms. Programming style. Object-oriented programming. Web programming. May be taught full term or 8 week. May be repeated for credit up to 3 hrs. Credit not given for both A598 and CSCI-A 202.

B403 Introduction to Algorithm Design and Analysis (3 cr.) Credit not given for both B403 and B503.

P423 Compilers (4 cr.) Credit not given for both P423 and P523.

P436 Introduction to Operating Systems (4 cr.) Credit not given for both P436 and P536.

P438 Fundamentals of Computer Networks (3 cr.) Credit not given for both P438 and P538. Not applicable toward a major in computer science.

B441 Digital Design (4 cr.) Credit not given for both B441 and B541. Not applicable toward a major in computer science.

P442 Digital Systems (4 cr.) Credit not given for both P442 and P542. Not applicable toward a major in computer science.

B443 Introduction to Computer Architecture (3 cr.) Credit not given for both B443 and B543.

B501 Theory of Computing (3 cr.) P: C241. Deterministic and nondeterministic automata, regular expressions, pumping lemmas; context-free languages, parsing, pushdown automata, context-sensitive languages, LBA, LR(k) languages, closure and decidability of language classes. Turing machines, random access machines, grammars, general recursive functions, equivalence of computation models, universal machines, relative computing. Unsolvability, semi-recursive sets, Rice's Theorem. Space and time complexity, NP completeness.

B502 Computational Complexity (3 cr.) Study of computational complexity classes, their intrinsic properties, and relations between them. Topics include time and space computational complexity, reducibility and completeness of problems within complexity classes, complexity of optimization problems, complexity hierarchies, relativization of the P=?NP conjecture, and parallel computation models and the class NC.

B503 Algorithms Design and Analysis (3 cr.) P: MATH M216, and C343. Models, algorithms, recurrences, summations, growth rates. Probabilistic tools, upper and lower bounds; worst-case and average-case analysis, amortized analysis, dynamization. Comparison-based algorithms: search, selection, sorting, hashing. Information extraction algorithms (graphs, databases). Graphs algorithms: spanning trees, shortest paths, connectivity, depth—first search, breadth—first search.

B510 Introduction to Applied Logic (3 cr.) Structures: relations between structures, term structures. Description: notation and meaning, substitution operations, first order formulas, database languages, program verification conditions, semantics valuation, normal forms, quantifier reduction, axiomatic theories. Proof: resolution, sequential calculi, natural deduction, automated theorem proving, semantic completeness. Limits of formalization: compactness, undecidability of truth, undecidability of canonical theories, nonformalizability of database theory.

P515 Specification and Verification (3 cr.) P: C311. Tools and techniques for rigorous reasoning about software and digital hardware. Safety, reliability, security, and other design-critical applications. Decision algorithms. Projects involving the use of automated reasoning, such as model checkers, theorem provers, and program transformation. Credit not given for both P415 and P515.

B521 Programming Language Principles (3 cr.) Systematic approach to programming languages. Relationships among languages, properties and features of languages, the computer environment necessary to support language execution.

B522 Programming Language Foundations (3 cr.) P: C311 or B521, and B510. Introduction to denotational, operational, and axiomatic approaches to programming language semantics. Semantic analysis of major programming language features. Logics of programs.

P523 Programming Language Implementation (3 cr.) P: B521 or C311. Implementation of traditional and nontraditional computer programming languages. Compilation, including lexical analysis, parsing, optimization, code generation, and testing. Run-time support, including run-time libraries, storage management, input-output. Comparison of implementation techniques. Extensive laboratory exercises.

B524 Parallelism in Programming Languages and Systems (3 cr.) P: P436 or P536, and either C311, H311 or B521, C343 or H343. Fundamentals of parallel computation, with an emphasis on parallel programming methodology and programming languages. Topics include: parallel algorithms; major paradigms for parallel software construction; (data parallelism, task/thread parallelism and CSP); compiling programs for parallel computers.

B534 Distributed Systems (3 cr.) A balanced treatment of fundamentals and practice of distributed systems. The foundational models, algorithms, and principles upon which distributed systems are based are studied in detail. These fundamentals are placed in the context of practical implementations by means of reading and critical analysis of research papers.

P535 Pervasive Computing (3 cr.) P: Object oriented programming. Topics in pervasive computing, such as sensors, mobility, tangibles, ambient displays, middleware, location and context-awareness; user-centered design methods, such as requirements gathering, design, prototyping, and evaluation. Labs cover current technologies, such as sensors and mobile devices. Lecture and laboratory. Lab fee.

P536 Advanced Operating Systems (3 cr.) P: C335 and C343, or honors versions. Advanced topics in operating systems, such as: multitasking, synchronization mechanisms, distributed system architecture, client-server models, distributed mutual exclusion and concurrency control, agreement protocols, load balancing, failure recovery, fault tolerance, cryptography, multiprocessor operating systems.

P538 Computer Networks (3 cr.) P: Operating systems or networking course. Layered TCP/IP architecture. LAN technologies (Ethernet, wireless, token rings). Switching. Internet addressing (IPv4, IPv6). Routing protocols. Congestion control (TCP, UDP). Applications (DNS, HTTP, peer-to-peer networks). Selection of topics including DHCP, ICMP, VPNs, multicast, security. Credit given for only one of P438 and P538.

B541 Hardware System Design I (3 cr.) P: C335 or honors version. Structured approach to hardware design,, emphasizing hardwired and microprogrammed control. Boolean algebra, hardware building blocks, architecture and control, implementation issues. In the laboratory, students build a working computer using hardware prototyping technologies. Basic training in the use of design and simulation software. Lecture and laboratory.

P542 Hardware System Design II (3 cr.) P: B541 or B441. Depending on instructor, a selection of topics in system-level design, such as simulation, logic synthesis, high-level synthesis, codesign, embedded software, verification, test, requirements specification, and others. Projects in system-level design. Computer-aided design tools. Lecture and laboratory.

B543 Computer Architecture (3 cr.) P: C335 and C343 or honors versions. Fundamentals of computer design, instruction processing and performance analysis. Architecture of single-processor systems, focusing on pipelining, memory and memory hierarchies, and interconnect technology. Exploration of architecture classes such as high-performance multiprocessors, massively parallel computers, embedded systems.

P545 Embedded and Real-Time Systems (3 cr.) P: Any 400-level "systems" course (middle digit 3 or 4). Design and implementation of purpose-specific, locally distributed software systems. Models and methods for time-critical applications. Real-time operating systems. Testing, validation, and verification. Safety-critical design. Related topics, such as resiliency, synchronization, sensor fusion, etc. Lecture and laboratory.

B551 Elements of Artificial Intelligence (3 cr.) P: C343 or H343, good knowledge of LISP or Scheme. Introduction to major issues and approaches in artificial intelligence. Principles of reactive, goal-based, and utility-based agents. Problem-solving and search. Knowledge representation and design of representational vocabularies. Inference and theorem proving, reasoning under uncertainty, planning. Overview of machine learning.

B552 Knowledge Based Artificial Intelligence (3 cr.) P: B551. Knowledge-based methods for artificial intelligence systems: knowledge representation, organization, and application. Typical content includes principles of memory organization, indexing and retrieval. Memory-based, analogical, and case-based reasoning. Applications to understanding, explanation, planning, and advisory systems.

B553 Neural and Genetic Approaches to Artificial Intelligence (3 cr.) P: CSCI-B 551. Approaches to the design of intelligent systems inspired by nervous systems, evolution, and animal behavior. Distributed and perceptually-grounded representations. Temporal processing. Perception and action. Genetic search. Unsupervised and reinforcement learning. Comparison of symbolic, subsymbolic, and hybrid approaches to intelligence.

B561 Advanced Database Concepts (3 cr.) P: C241, C335, and C343 or honors versions. Database models and systems, especially relational and object-oriented; relational database design theory; structures for efficient data access; query languages and processing; database applications development; views. Transaction management: concurrency and recovery.

P565-P566 Software Engineering I-II (3-3 cr.) P: C343, B461 previously or B561 concurrently. Analysis, design, and implementation of software systems. Requirements specification: data and process modeling. Software design methodologies. Software quality assurance: testing and verification. Software development processes.

P573 Scientific Computing (3 cr.) P: MATH M303 or M301, M343, and C212 or H212. For students from all scientific, engineering, and mathematical disciplines, this course provides an overview of computer hardware, software, and numerical methods that are useful on scientific workstations and supercomputers. Topics include high-performance computer architectures, software tools and packages, characteristics of numerical methods in common use, graphical presentation of results, and performance analysis and improvement.

B581 Advanced Computer Graphics (3 cr.) P: C343, MATH M301 or M303, or equivalent experience. Introduction to graphics hardware and software. Two-dimensional graphics methods, transformations, and interactive methods. Three-dimensional graphics, transformations, viewing geometry, object modeling and interactive manipulation methods. Basic lighting and shading. Video and animation methods.

B582 Image Synthesis (3 cr.) P: B581, MATH M215. Raster image display: color theory, gamma correction, and filtering. Advanced shading methods: local illumination models, global illumination models. Surface display, including ray tracing and Z-buffering. Solid modeling: spline surfaces, CSG, superquadrics, and deformations. Scientific visualization: isosurfaces and volume rendering.

B599 Teaching in Computer Science (1 cr.) General principles of teaching and practical experiences that relate to teaching computer science. An important feature of the course is the microteaching, in which each participant prepares and delivers short lectures to the seminar participants. Each presentation is followed by critical analysis and discussion.

B603 Advanced Algorithms Analysis (3 cr.) P: B503. Advanced topics in analysis of algorithms, including fast algorithms for classical problems, lower bounds results, and statistical behavior.

B607 Philosophy of Computation (3 cr.) P: Consent of the instructor. Critical examination of the conceptual foundations of computing. Several different views assessed with respect to conceptual, explanatory, and empirical criteria. Primary focus on formal symbol manipulation, recursive function theory, effective computability, computational complexity, digitality, and information processing. Some nonstandard approaches also considered: connectionism, dynamics, and artificial life.

B609 Topics in Algorithms and Computing Theory (1-6 cr.) P: Instructor's permission. Special topics in algorithms and computing theory. May be repeated for credit with permission.

B619 Topics in Applied Logic (1-6 cr.) P: Instructor's permission. Special topics in applied logic. May be repeated for credit with permission.

B621 Advanced Concepts in Programming Languages (3 cr.) P: Either C311, H311, or B521. P or C: P423 or P523. Discussion of current issues in the design of programming languages. Modularity, abstraction, and static analysis. Applicative and nonapplicative models. Single and multiple processing.

B622 Programming Language Type Systems (3 cr.) P: C311 or B521. Theoretical foundations and engineering techniques for modern type systems, focusing on polymorphism and subtyping in typed lambda-calculi; applications, including type systems for objects, abstract data types, and modules; issues in type checker implementation and polymorphic type inference.

B629 Topics in Programming Languages (1-6 cr.) P: C311 or B521 and instructor's permission. Special topics in programming languages. May be repeated for credit with permission.

B639 Topics in Software Systems (1-6 cr.) Special topics in software systems. May be repeated for credit with permission.

B644 Very Large Scale Integration (3 cr.) P: B441 or B541. Basic theory and practice required to convert hardware algorithms and architecture to silicon structures. Use of state-of-the-art design tools for integrated circuits. Lab fee.

B649 Topics in Systems (1-6 cr.) P: Instructor's permission. Special topics in systems. May be repeated for credit with permission.

B651 Natural Language Processing (3 cr.) P: B551. R: B552 or B553. Theory and methods for natural language processing. Algorithms for sentence parsing and generation. Context-free and unification grammars. Question-and-answer systems. Analysis of narratives. Finite-state approaches to computational phonology and morphology. Machine translation. Machine learning of natural language. Speech recognition. Neural-network and statistical alternatives to symbolic approaches.

B652 Computer Models of Symbolic Learning (3 cr.) P: B552. Symbolic artificial intelligence methods for learning. Inductive and explanation-based generalization. Failure-driven learning. Case-based learning. Typical content includes operationality of explanations and utility of learning. Goal-driven learning. Criteria for when, what, and how to learn. Learning in integrated architectures.

B656 Web Mining (3 cr.) Machine learning techniques to mine the Web and other unstructured/semistructured, hypertextual, distributed information repositories. Crawling, indexing, ranking and filtering algorithms using text and link analysis. Applications to search, classification, tracking, monitoring, and Web intelligence. Group project on one of the topics covered in class.

B657 Computer Vision (3 cr.) P: C463 or B551. Concepts and methods of machine vision as a branch of artificial intelligence. Basics of digital image processing. Local and global tools for deriving information from image data. Model-based object recognition and scene understanding.

B659 Topics in Artificial Intelligence (1-6 cr.) P: Instructor's permission. Special topics in artificial intelligence. May be repeated for credit with permission.

B661 Database Theory and Systems Design (3 cr.) P: B461 or B561. Database models: relational, deductive, complex-object, object-oriented. Query languages: relational algebra and calculus, datalog, fixpoint logics, object-oriented query languages. Transaction management theory: concurrency control, recovery, distribution. Post-relational and object-oriented database systems.

B665 Software Engineering Management I (3 cr.) P: B561 or BUS S560. Topics include the high cost of software, the software life cycle, understanding programming teams, and methodologies for controlling development. Presentation of readings and supervision of programming teams producing software products required.

B666 Software Management Implementation II (1-3 cr.) P: B665. Continuation of projects from B665. Periodic reports and a final paper required. If taken for two or more credits, an additional project or paper is required.

B669 Topics in Database and Information Systems (1-6 cr.) P: Instructor's permission. Special topics in database and information systems. May be repeated for credit with permission.

B673 Advanced Scientific Computing (3 cr.) P: P573 and MATH M471. Multiprocessor organization: vectorization, memory organization, processor topologies and architectures. Models of parallelism. Programming language and systems for scientific and high-performance computing. Environments for interactive scientific experiments and databases. Distributed programming tools. Parallelism in scientific problems: parallel algorithmic techniques, parallel algorithms and models, parallel performance analysis and debugging.

B679 Topics in Scientific Computing (1-6 cr.) P: Instructor's permission. Special topics in scientific computing. May be repeated for credit with permission.

B689 Topics in Graphics and Human Computer Interaction (1-6 cr.) P: Instructor's permission. Special topics in graphics and human computer interaction. May be repeated for credit with permission.

Y790 Graduate Independent Study (1-6 cr.) Independent study under the direction of a faculty member, culminating in a written report. May be repeated for credit. R grade not allowed. The different departmental options for independent study are: Research and Reading, Software System Development, Master's Research Project, Master's Software Project, and a University Master's Thesis.

Y798 Professional Practicum/Internship (non-credit) P: Current enrollment in graduate degree program in computer science. Provides for participation in graduate-level professional training and internship experience.

Y890 Thesis Readings and Research (1-12 cr.) Research under the direction of a member of the graduate faculty leading to a Ph.D. dissertation.