COMP-667
Software Fault Tolerance
Course Outline

Instructor

Jörg Kienzle, McConnell 327
Phone: (514) 398-2049, home (emergency): (514) 871-2780
Email: Joerg.Kienzle@mcgill.ca
Office hours: Monday: 11:30 - 12:30 + any other time (send email)

Teaching Assistant

Wisam Al Abed, McConnell 322
Phone: (514) 398-7071 ext. 00116
Email: Wisam.Alabed@mail.mcgill.ca
Office hours: TBA

Course Homepage

http://www.cs.mcgill.ca/~joerg/SEL/COMP-667_Home.html

Course Contents

The goal of this course is to study the techniques that can be applied by software developers to produce fault-tolerant software, e.g. software that continues to deliver service in spite of the operational effects of software design faults and faults of the surrounding environment. The course aims not only at presenting the concepts, but concentrates on implementation issues as well.

The first part of the course presents the need for reliability, and puts fault tolerance in relation with other reliability issues, e.g. verification and validation, fault prevention, quality assurance, etc. Then, the main concepts of fault tolerance are introduced: failure classification, types of redundancy, types of recovery. The features available in modern programming languages to support fault tolerance are reviewed: exceptions, serialization, threading.

The second part of the course concentrates on different forms of error recovery for sequential and concurrent systems. The notion of error confinement and the idealized fault-tolerant component are presented. Various advanced transaction and atomic action models are studied, including ways of implementing the schemes in programming languages. Advantages and disadvantages of the different models are highlighted.

The third part finally focusses on design diversity, e.g. N-version programming, and data diversity. Special emphasis is put on the decision mechanisms used with many of the seen techniques.

If time permits, the course will also present techniques that can be applied to standard model-driven software development processes to make them dependability-aware.

Tentative Course Schedule

- Overview & Questionnaire (1 lecture)
- Fundamental Concepts (2-3 lectures)
- Programming Language Features (3 lectures)
- Independent Concurrent Systems (1 lectures)
- Implementing N-version Programming (1 lectures)
- Sequential Techniques (1-2 lectures)
- Implementing Backward Error Recovery (2 lectures)
- Competitive Techniques (2 lectures)
- Implementing Transaction Support (3 lectures)
- Dependability-Focused Requirements Engineering (2 lectures)
- Cooperative Techniques (2 lectures)
- Hybrid Systems (2 lectures)

Reading List and Handouts

All course slides and exercises will be available for download on the course webpage.

Textbooks that could be Helpful

- Laura L. Pullum: Software Fault Tolerance Techniques and Implementation, Artech House 2001. (ISBN: 1580531377)

This book covers the introduction, as well as the part on design diversity and its implementation in detail. Unfortunately, transaction models, atomic actions and their implementation techniques are not covered at all. An electronic version is on the web at
http://www.books24x7.com/marc.asp?isbn=1580531377. Unfortunately, the free trial period for this site is only 1 week.

- Jörg Kienzle: Open Multithreaded Transactions - A Transaction Model for Concurrent Object-Oriented Programming, Kluwer Academic Publishers, 2003. (ISBN: 1-4020-1727-8)

This book covers transaction models, atomic actions and most of all open multithreaded transactions. The design of an object-oriented framework implementing transaction support is presented, and some parts of the implementation in Ada are shown. Examples of interfaces for procedural, object-oriented and aspect-oriented programming languages are presented.

- Lee, P. A.; Anderson, T.: Fault Tolerance - Principles and Practice, 2nd edition, Springer Verlag, 1990.

This book covers all parts of the course, but is a little outdated. In particular, the recent development in the field of advanced transaction and atomic action models are not addressed, and it does not go into implementation details.

- Ramamritham, K.; Chrysanthis, P. K.: Advances in Concurrency Control and Transaction Processing, ACM Press, Los Alamitos, California, 1997.

This book covers in detail different transaction models and concurrency control techniques employed in transaction processing.

- Jean-Claude Geffroy and Gilles Motet: Design of Dependable Computing Systems, Kluwer Academic Publishers, 2002. (ISBN 1-4020-0437-0)

This book does a very good job in presenting the fundamental concepts of fault tolerance. It also goes into detail on fault avoidance and fault removal.

Prerequisites

- COMP-409 Concurrent Programming or consent of instructor

Course Format

The course will be offered in the traditional lecture format, i.e. 3 hours of lectures per week.

Grading

There will be 1 early warm-up assignment (5%), 2 homework programming assignments (probably 2 * 20%), 1 non-programming assignment (10%), and a project (45%). The project consists of either

- implementing a software fault tolerance scheme (distributed or concurrent) as a library / framework for a programming language of your choice, or
- presenting a case study of a fault tolerant system / middleware in class

Note on Academic Integrity

McGill University values academic integrity. Therefore, all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures (see www.mcgill.ca/integrity for more information).