

Software Tools Research Papers

Part II - Lecture 2

Today's Overview

- Writing Research Papers
- Research Paper Assignment
- Research Paper Presentations

2009
YEAR

COMPSCI 732

The University of Auckland | New Zealand

Writing Research Papers

Research Papers

Typical structure of a research paper (or thesis):

1. **Title**
2. **Abstract:** brief summary, typically 100-200 words
3. **Introduction:** motivation, context, overview
4. **Related Work:** what are others doing
5. **Design / Concepts:** the theory / our new ideas
6. **Implementation:** how we did it
7. **Evaluation:** why it is good/useful/better than others
8. **Conclusion:** summing up the results

Short conference papers: typically around 4 pages

Long conference papers: typically about 10 pages

Paper Writing Strategy

1. **Related Work:** get an overview and note down points
2. **Design / Concepts:** collect ideas
 1. Create structure with bullet points / mind map
 2. Create figures
3. **Implementation:** create a prototype
 1. Start small and extend it bit by bit
 2. Experiment and collect results
(more bullet points and other data)
4. **Evaluation:** compare and refine your work
(if necessary, go back to 2 or 3)
5. **Title, Abstract, Introduction and Summary** can be done last

Tackling Related Work

1. Gather phase

- Keyword search
(e.g. Google Scholar, ACM, IEEE)
- Follow up the references
(cited and citing papers)

2. Filter phase:

read only abstract and throw blanks out

3. Reading phase

The "someone else has already done it" problem

- Look again, is it really the same?
- Related work is good!



Research Paper Assignment

Paper Topics

1. Static Analysis
2. Program Exploration
3. IDEs
4. Tool Integration
5. Aspects
6. Software Engineering
7. Web Development



- About 5 papers per topic
- 2 persons per paper (presentation is done together) except for papers marked as "short" (one person)
- If you find another interesting paper you would prefer to present, ask Christof about it

Topic: Static Analysis

How to analyze programs by looking at their code?

- Pär Emanuelsson, Ulf Nilsson. *A Comparative Study of Industrial Static Analysis Tools*. SSV, 2008.
- William R. Bush, Jonathan D. Pincus, David J. Sielaff. *A static analyzer for finding dynamic programming errors*. *Software: Practice and Experience*, 2000.
- Bacon, D.F. and Sweeney, P.F. *Fast static analysis of C++ virtual function calls*. *OOPSLA*, 1996.
- Lattner, Adve. *LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation*. *GCO*, 2004.
- Leroy. *Java Bytecode Verification: An Overview*. *CAV*, 2001.

Topic:

Program Exploration

How can we understand code faster?

- Emily Hill, Lori Pollock, and K. Vijay-Shanker. Exploring the Neighborhood with Dora to Expedite Software Maintenance. ASE, 2007.
- Martin P. Robillard, Wesley Coelho, and Gail C. Murphy. How Effective Developers Investigate Source Code: An Exploratory Study. TSE, 2004.
- Mik Kersten, Gail C. Murphy. Mylar: a degree-of-interest model for IDEs. AOSD, 2005.
- Martin P. Robillard. Automatic Generation of Suggestions for Program Investigation. FSE, 2005.
- Andrew J. Ko and Brad A. Myers. Debugging Reinvented: Asking and Answering Why and Why Not Questions about Program Behavior. ICSE, 2008.

Topic: IDEs

How can we make IDEs better tools?

- B. Medeiros. Creating IDEs for the Eclipse Platform. Tech Report, 2007.
- A. D. Eisenberg, G. Kiczales. Expressive Programs Through Presentation Extension. AOSD, 2007.
- V. Gruhn. Process-Centered Software Engineering Environments. Annals of SE, 2002.
- W. Harrison, H. Ossher, P. Tarr. Soft. Eng. Tools and Environments: A Roadmap. ICSE, 2000.
- Murphy et al. How Are Java Software Developers Using the Eclipse IDE? IEEE Software, 2006. (**short**)
- Hupfer et al. Introducing Collaboration into an Application Dev. Env. CSCW, 2004. (**short**)

Topic: Tool Integration

2009
YEAR

COMPSCI 732

The University of Auckland | New Zealand

How can we integrate different tools and models?

- Reichmann et al. *GeneralStore - A CASE Tool Integration Platform*. ECBS, 2004.
 - Kappel et al. *On Models and Ontologies*. MOD, 2006.
 - M.N. Wicks, R.G. Dewar. *A new research agenda for tool integration*. JSS, 2007.
 - L. Tratt. *Model transformations and tool integration*. SSM, 2005.
 - Königs, Schürr. *Multi-Domain Integration with MOF and extended Triple Graph Grammars*. Dagstuhl, 2005.
 - Burmester et al. *Tool Integration at the MetaModel Level within the FUJABA Tool Suite*. TIS, 2003.
- (short)**

Topic: Aspects

2009
YEAR

COMPSCI 732

The University of Auckland | New Zealand

How to deal with crosscutting concerns?

- Kiczales et al. *Getting started with AspectJ*. ComACM, 2001.
- G. Kiczales, M. Mezini. *Aspect-Oriented Programming and Modular Reasoning*. ICSE, 2005.
- Tarr et al. *N Degrees of Separation: Multi-Dimensional Separation of Concerns*. ICSE, 1999.
- D. Janzen, K. De Volder. *Navigating and Querying Code Without Getting Lost*. AOSD, 2003.
- J. Pfeiffer, A. Sardos, J. R. Gurd. *Complex Code Querying and Navigation for AspectJ*. Eclipse, 2005. **(short)**

Topic:

Software Engineering

How to develop maintain and deploy software?

- Alan Dearle. Software Deployment, Past, Present and Future. FOSE, 2007.
- R. Robbes, M. Lanza. A Change-based Approach to Software Evolution. ENTCS, 2007.
- Müller et al. Reverse Engineering: A Roadmap. ICSE, 2000.
- T. Mens, T. Tourwe. A Survey of Software Refactoring. TSE, 2004.
- J. Greenfield, K. Short. Software Factories - Assembling Applications with Patterns, Models, Frameworks and Tools. OOPSLA, 2003.

Topic: Web Development

2009
YEAR

COMPSCI 732

The University of Auckland | New Zealand

How to develop Web applications efficiently?

- D. Draheim, and G. Weber. Specification and Generation of Model 2 Web Interfaces. *APCHI*, 2004.
- Draheim et al. Realistic Load Testing of Web Applications. *CSMR*, 2006.
- Draheim, Lutteroth, Weber. A Source Code Independent Reverse Engineering Tool for Dynamic Web Sites. *CSMR*, 2005.
- P. Tonella, F. Ricca. Dynamic Model Extraction and Statistical Analysis of Web Applications. *WSE*, 2002.
- D. Draheim, E. Fehr, G. Weber. JSPick - A Server Pages Design Recovery Tool. *CSMR*, 2003. **(short)**

Research Paper Group Work

1. Find a partner
2. Pick a research paper
3. Read your paper together
4. Discuss the paper with other groups (with similar papers)



2009

YEAR

COMPSCI 732

The University of Auckland | New Zealand

Research Paper Presentations



Presentations

2009
YEAR

COMPSCI 732

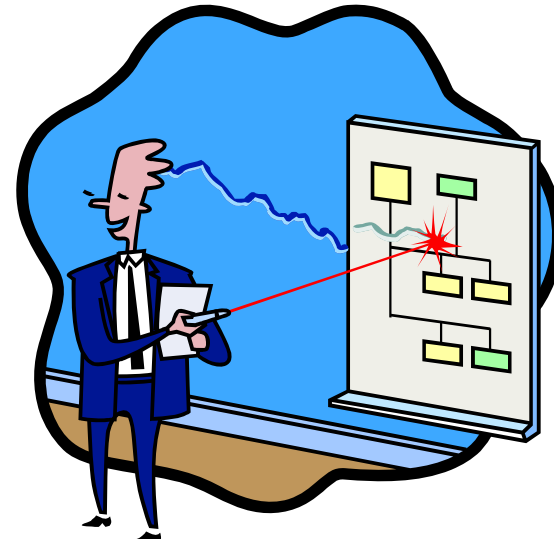
The University of Auckland | New Zealand

Normal papers

- In teams of two, 4 minutes each
- 2 minutes questions

Short papers

- One person, 4 minutes only
- 1 minute questions
- Should explain context, refer to some related work
- You don't have to preset full paper, only main ideas
- You may use illustrations from the paper



Presentation Grading Schedule

- Are the important results/concepts clearly explained?
- Are the important results/concepts clearly presented on the slides?
- Is related work mentioned (briefly)?
- Are questions adequately answered?



Tips:

- Do a "dry run" of your presentation before you give it before the class
- Structure and practice is everything!