CISC883: LECTURE 2 INTRODUCTION TO ULSS

Cor-Paul Bezemer

Website

http://sailhome.cs.queensu.ca/~corpaul/cisc883_2016

Course Deliverables

- ULSS Design Position Paper
 - 2-4 page position paper discussing how one ULSS system of choice matches Lampson's design principles
 - Due: Wednesday, 19 October, 2:30 pm

Course Deliverables

- Weekly Modules (7)
 - Weekly paper recommendation
 - Weekly paper review from previous week
 - Two 45-minute individual in-class presentations about a topic
 - Send me your preferred topics!

Characteristics of ULSSs

- 1. Decentralization
- 2. Inherently conflicting, unknowable, & diverse requirements
- 3. Continuous evolution & deployment
- 4. Heterogeneous, inconsistent, & changing elements
- 5. Erosion of the people/system boundary
- 6. Normal failures
- 7. New paradigms for acquisition & policy

We discuss:

- Each characteristic
- The assumptions that the characteristic undermines

Characteristics of ULSSs (1)

Decentralization

Undermined Assumption(s):
All conflicts must be resolved, & resolved centrally & uniformly

Characteristics of ULSSs (2)

Inherently conflicting, unknowable, & diverse requirements

- Undermined Assumption(s):
- Requirements can be known in advance & change slowly as experience with a system grows
- Trade-off decisions will be stable

Characteristics of ULSSs (3)

Continuous evolution & deployment Undermined Assumption(s): • System improvements are introduced at discrete intervals (build-use-build)

Characteristics of ULSSs (4)

Heterogeneous, inconsistent, & changing elements

- Undermined Assumption(s):
- The effect of a change can be predicted sufficiently well.
- Configuration information is accurate & can be tightly controlled.
- Components & users are fairly homogeneous.

Characteristics of ULSSs (5)

- Erosion of the people/system boundary
- Undermined Assumption(s):
- People are just users of the system.
- The collective behavior of people is not of interest.
- Social interactions are not relevant.

Characteristics of ULSSs (6)

Normal failures

- Undermined Assumption(s):
- Failures will occur infrequently.
- Defects can be removed.

Characteristics of ULSSs (7)

New Paradigms for Acquisition & Policy

Undermined Assumption(s):

 A prime contractor is responsible for system development, operation, & evolution.



Software Design Principles

Software Design is the study of methods, technologies, languages, principles & practices that make it possible to create, validate & evolve complex software systems.



Software Design Principles

- Hints for Computer System Design, Lampson (1983)
- Hints on system design, based on experience with systems & SOSs
 - Experience includes successes & failures, what worked & what didn't
- Do these carry over to ULSSs?

Software Design Principles

Hints on how to build a "good" system, with respect to:

1. Functionality

 How to get a system to do the things you want it to do

2. Speed

How to make a system faster

3. Fault Tolerance

• How to make a system reliable

Functionality

- Keep it simple
 - Do one thing at a time, & do it well
 - Don't generalize
- Make it fast, rather than general or powerful
- Use procedure arguments to provide flexibility in the software
- Leave it to the client

Functionality

- Unix pipe command
 - Build small programs that take one or more character streams as input, produce one or more streams as output, & do one operation
 - Is -al | grep myfile

Functionality

- Trade off: Improve a design Vs. stability
 - Keep basic interfaces stable
 - Keep a place to stand, if an interface must be changed
- Divide & Conquer
- Handle normal & worst cases separately

Speed

- Split resources in a fixed way if in doubt, rather than sharing them
- Use static analysis if you can
- Cache answers to expensive computations, rather than doing them over

Speed

- memcached
 - General purpose distributed memory caching system
 - Intended to speed up dynamic web applications
 - RAM store of (key,value) pairs to reduce database access
 - Stores results of, e.g., database or API calls
 - Similar to a database, with fast read/write

Speed

- When in doubt, use brute force
- Compute in background, when possible
- Use batch processing
- Safety first
- Shed load to control demand, rather than allowing the system to become overloaded.

Fault Tolerance

- End-to-end
- Log updates
- Make actions atomic or restartable

Assignment

- Due Wednesday, 19 October, 2:30 pm
 - 2-4 page position paper discussing how one ULSS system of choice matches Lampson's design principles
 - IEEE style (Latex!)

CISC883: LECTURE 2 RESEARCH METHODS

Writing Technical Papers & The Task of the Referee

Writing Technical Papers

- Research Papers Vs. Survey Papers
- The breakdown of good research papers

 "Writing Technical Articles": http://www.cs.columbia.edu/~hgs/etc/writin g-style.html

Writing Technical Papers

- A good research paper has:
 - A clear statement of the problem the paper is addressing
 - The proposed solution(s)
 - The results achieved
- A good research paper describes:
 - What has been done before on the problem
 - What is new

Writing Technical Papers

A paper's goal is to describe novel technical results. 1.An algorithm

2.A construct, e.g., software system, protocol

3.A performance evaluation, e.g., analysis, simulation 4.A Proof or a theory (a collection of theorems)

A research paper should focus on:

- Describing the results in details to establish their validity
- Identifying the novel aspects of the results
- Identifying the significance of the results

Writing Technical Papers: *Paper Structure*

- Abstract (100-150 words)
- Introduction (brief!)
- Related Work (or before conclusion)
- Outline of the rest of the paper
- Body of paper
 - Problem, {approach, architecture, realization, results
- Conclusion and Future Work
- Acknowledgements
- Bibliography
- Appendix

DiscussioDiSection Then Related Work

Writing Technical Papers: Order

- Abstract (100-150 words)
- Introduction (brief!)

5

3

- Related Work (or before summary)
- Outline of the rest of the paper
- Body of paper
 - Problem
 - {approach,architecture,realization}
- results
- Summary and Future Work
- Acknowledgements
 - Bibliography
 - Appendix



Writing Technical Papers: *Title & Authors*

- Title
 - Avoid abbreviations (less the well known ones)
 - No false promises
- Authors:
 - Made a significant contribution to the research underlying the paper (e.g., experimental design, prototype development)
 - Contributed to writing/reviewing the paper
 - Approved the final version of the paper

Writing Technical Papers: Abstract

- Typically not more than 100-150 words
- Highlight the problem & the principal results
- Abstract will be used by search engines; must have terms that identify your work
- Avoid math, general motivation, "in this paper"
 No citations; abstract can be used without the main paper

Writing Technical Papers: *Introduction*

- Tell the reader what this paper is about, not just how important your research area is.
- Brief introduction to problem, outline of the proposed solution & contribution
- Set up the expectations for the rest of the paper
 - Provide context and a preview
- Don't repeat the abstract in the intro!

Writing Technical Papers: *Related Work*

- Brief survey of the area
- Include recent & relevant results
 - Cite the work of the PC co-chairs & as many other PC members as possible
 - Cite relevant work from previous conference proceedings or journal volumes

Writing Technical Papers: *Paper Outline*

 "The remainder of the paper is organized as follows. In Section 2, we introduce ...Section 3 describes ... Finally, we describe future work in Section 5."

Writing Technical Papers: **Body of Paper**

- State & Motivate the problem: include example scenarios
- Describe
 - Approach
 - Architecture: more generic than the implementation
 - Realization: implementation details
 - Used language, platform, dependencies
- Evaluation: Demonstrate results
 - How does it really work in practice?

Writing Technical Papers: **Body of Paper** Evaluation: Demonstrate results

- Report results & simulations in detail: reader can duplicate the results.
 - Parameters used, # of samples, initial conditions
- Discuss statistical confidence, confidence intervals
- Choose graphs wisely & justify strange behavior in them
 Flat graphs are not entertaining !

Writing Technical Papers: **Discussion (Optional)**

- Discuss useful insights in your approach or in the results
 - Strengths and weaknesses
 - Threats to Validity
 - Comparison between results of conducted case studies

Writing Technical Papers: Summary & Future Work

- Clearly & concisely summarize the work
 Often repeats the main result
- Future work provides other avenues for research in this topic
 - Presumably this paper did not solve all of the problems

Writing Technical Papers: Acknowledgements

- Acknowledge your funding sources, e.g., "This work was supported in part by XYZ under grant ABC."
- Don't acknowledge anonymous reviewers, unless they provided an exceptional feedback.

Writing Technical Papers: **Bibliography**

- Book citations include publication years, but no ISBN number.
- Can include URL to material, e.g., downloadable software
- Make sure to cite the source, date & other identifying information

Writing Technical Papers: Appendix

- Includes low-level, important details, e.g.,
 - Detailed protocol descriptions
 - Proofs with more than two lines
 - Code snippets
- To be cut out first if forced to

Writing Technical Papers: Conference Reviewing Process

- Paper is submitted to the program chairs
- Program chair assigns a paper to one or more reviewers (committee members)
- Reviewer may provide the review or delegate to a trusted source
- Online discussion by reviewers
- Reviews are sorted by score, & the "best" papers are accepted for publication

CISC883: LECTURE 2 RESEARCH METHODS

Writing Technical Papers & The Task of the Referee

What is peer reviewing?

- Papers are submitted for publication to conferences, journals, newsletters, etc.
- Such publications use referees as external experts to evaluate papers.
- This process is called *peer review*.

• A public service: a professional obligation of a computer science & engineering professional.

Task of the Referee

- Evaluate a paper for publication in a journal or conference proceedings
- Determine:
 - If the work is correct
 - If the problem & results are new & significant
 - If the presentation is high quality
 - What changes to the paper are necessary/desirable
- Evaluation must be with regard to the coverage & degree of selectivity of the publication.

Publishable

- A publishable paper: makes a sufficient contribution
 - New & interesting research
 - New & insightful synthesis of existing results
 - A useful survey or tutorial
- Referee: provides an opinion as to whether the paper makes a sufficient contribution
- No single correct evaluation of the paper

Task of the Referee

- Two major deliverables in a referee report
 - A recommendation for or against publication or presentation
 - A list of necessary & recommended changes
 Revisions for Conferences Vs. Journals

Reading a Paper

- Refereeing is different than reading published work
 - Analogous to a professor grading a paper
 - No assumption that the work is correct, novel, or worthwhile
- Paper should be read with an open mind with the goal of delivering a referee report
- Badly flawed papers ...

- A referee's opinion about whether or not the paper should be published
- Used by the editor or program chair
- Read by the authors to improve their work

- Brief recommendation and reasons for it
- Brief summary of paper (1-5 sentences)
- Evaluation of the goal & quality of the work
- Overall recommendation as to publication
 - Strength of recommendation
 - Reject, Weak Reject, Weak Accept, Accept
 - Confidence of Reviewer
 - How qualified are you to evaluate this work?

- Typically, a referee is anonymous
- You are expected to be reasonable
 - Clear & explicit constructive criticism
 - Words such as "fool" & "idiot" are bad, terms such as "trash" are not appropriate

- Reports must be returned in a timely fashion
 - Long delays between submission & publication due to referees & publication queue
 - Workshops, Conferences, Journals

Evaluating Research

- What is the purpose of the paper?
- Is this paper appropriate?
- Is the goal of this paper significant?
- Is the method of approach valid?

Evaluating Research

- Is the actual execution of the research correct?
- Are the correct conclusions being drawn from the results?
- Is the presentation satisfactory?
- What did you learn?

Other Issues

- Simultaneous Submission
- Prior Publication
- Unrevised Retries

Acknowledgements

- Contributions of others must be fully acknowledged
- Papers (clearly) should not plagiarize the work of others
 - Not just entire papers, but also subsections, e.g., Related Work

Presentations

- Large-Scale Applications
- Web APIs & Web Services
- Hosted Applications
- Infrastructure for Rent
- Autonomic computing & monitoring platforms & approaches
- Mobile App Platforms
- Tools for Large-Scale Analysis

Assignment

- Due Wednesday, 19 October, 2:30 pm
 - 2-4 page position paper discussing how one ULSS system of choice matches Lampson's design principles
 - IEEE style (Latex!)