Analyzing and Testing Non-functional Software Properties

Sudipta Chattopadhyay
Kick-off Meeting

Objective

- Learn about non-functional software properties

  - Timing

  - Energy

  - Robustness

  - Security (primarily side-channel leaks)

  - Reliability

End of the seminar

- Answers to questions you should know (or have some ideas at least)

  - Why should we care about non-functional properties or should we?

  - Is it harder to analyse than functional properties?

  - Is it harder to test than functional properties?

  - What is the formal definition for non-functional bugs?

  - Which applications are more critical if they are prone to non-functional bugs?

  - Are non-functional bugs dependent on execution platforms (processors, memory) or are they independent?

  - Are you kidding me? So how do developers fix non-functional bugs or they just ignore them?

End of the seminar

- Answers to questions you should know

  - What happens if I blindly apply a technique developed for analysing/testing functional properties?

  - What does performance/energy debugging mean to you?

    - Does it differ from classic debugging? Is it harder than classic debugging?

  - Isn't it a compiler's job? Why should I care about?

  - Meh, Sudipta is bluffing….People must have looked at it in the past. Have they? If not, why not?

  - Should we care about this problem for next 5 years?

    - Can I work on it for my PhD thesis **now**?

  - Should we care about this problem for next 10 years?

    - Can I work on it for my PhD thesis after **5 years** (I have no time to do PhD now)?

Organizational matters

- No unexcused absence is accepted

  - You need to give a valid reason in advance (unless it is accidental)

- Submit all summaries by Wednesday each week

Organizational matters

- Final presentation

  - Need to present a paper at the end of the semester

  - It's not just the presentation of the paper

  - Study related literature

  - We should get a better understanding of the overall area

Reading papers

- Read every detail

  - possibly some related paper if needed

- Understand the main concept

- Ask questions

  - Is this contribution fundamental or incremental?

    - General idea vs. simple improvement over state-of-the-art

  - Is the proposed claim justified?

  - Is this contribution theoretical with no hope of being applied?

  - Is this pure engineering?

  - Is this just one-off paper or there could be a series of follow-up papers?

  - Can I write a paper in this area now (or technology has overshadowed the contribution)?

Writing summaries

- Summary is not a short version of the paper

- Do not choose sentences from the paper

- Do not strictly follow the flow of the paper (Section 1 mentions this, …..)

- Do not write in first person to refer the work (Our work addressed….)

- Not providing citations from another paper/blog/website

- Highlight

    - Problem addressed

    - Main idea

    - Contribution

    - Weaknesses

    - **Evaluation**

    - Questions highlighting the potential ambiguity of the technique, evaluation or the paper at large

Writing summaries

- DART: Directed Automated Random Testing

- *"In Section 5, authors have introduced their technique."*

Writing summaries

- *"In Section 5, authors have introduced their technique."*

- This sentence does not convey anything

  - Which technique?

  - What problem?

  - Who cares about section 5?

- The paper addresses the problem of testing software functionality bugs via dynamic symbolic execution.

Writing summaries

- *"The paper addresses the problem of testing software functionality bugs via dynamic symbolic execution. The proposed technique is extraordinary."*

Writing summaries

- *"The paper addresses the problem of testing software functionality bugs via dynamic symbolic execution. The proposed technique is extraordinary."*

- Do not use flashy words

  - extraordinary

  - unbelievable

  - genius

  - great

  - poor, bad, pointless, I don't care

Writing summaries

- *"The paper addresses the problem of testing software functionality bugs via dynamic symbolic execution. Although it was really hard for me to understand, the proposed technique seems to work well."*

Writing summaries

- *"The paper addresses the problem of testing software functionality bugs via dynamic symbolic execution. Although it was really hard for me to understand, the proposed technique seems to work well."*

- Do not express your emotion

  - hard for me, easy for me

  - I enjoyed a lot reading it

The Seminar

- Divided into categories

  - Analysis/Verification
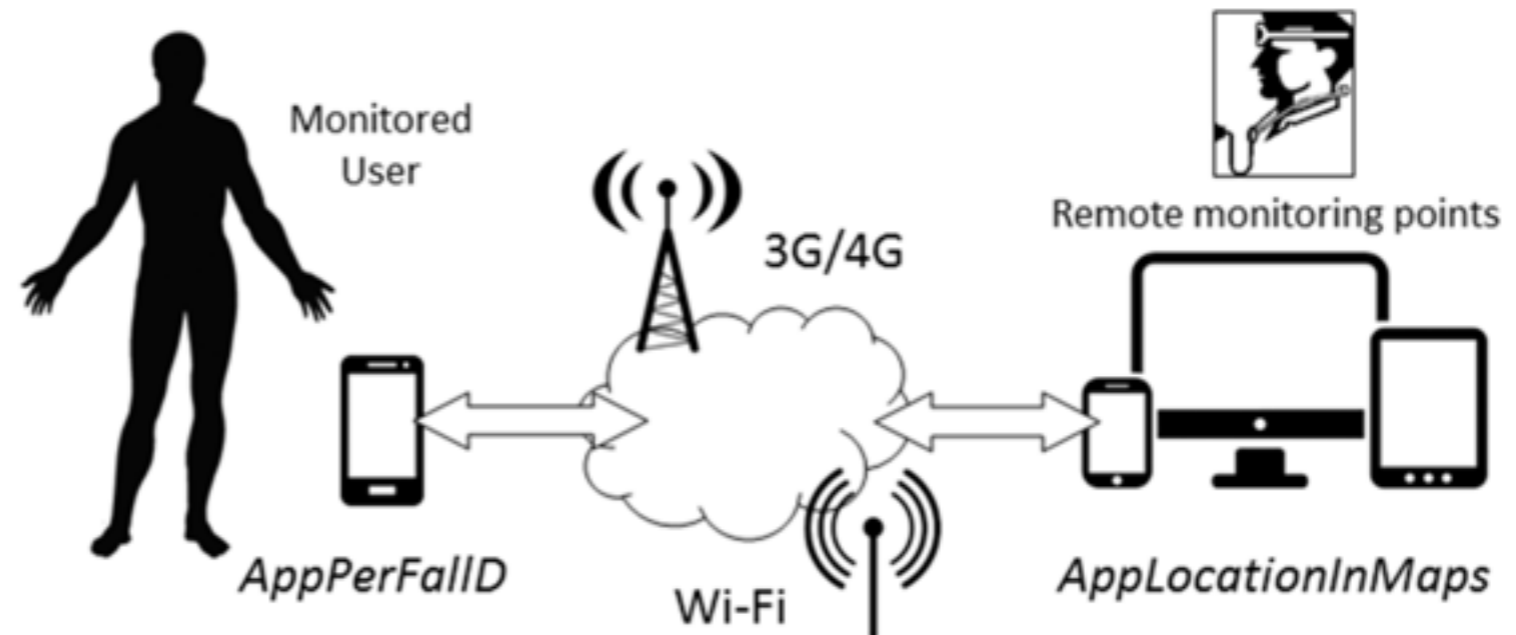
  - Testing

  - Profiling/Mining

The Seminar

- Performance/energy bugs

  - Waste performance or energy

    - Unnecessary memory access

    - Unnecessary GPS access

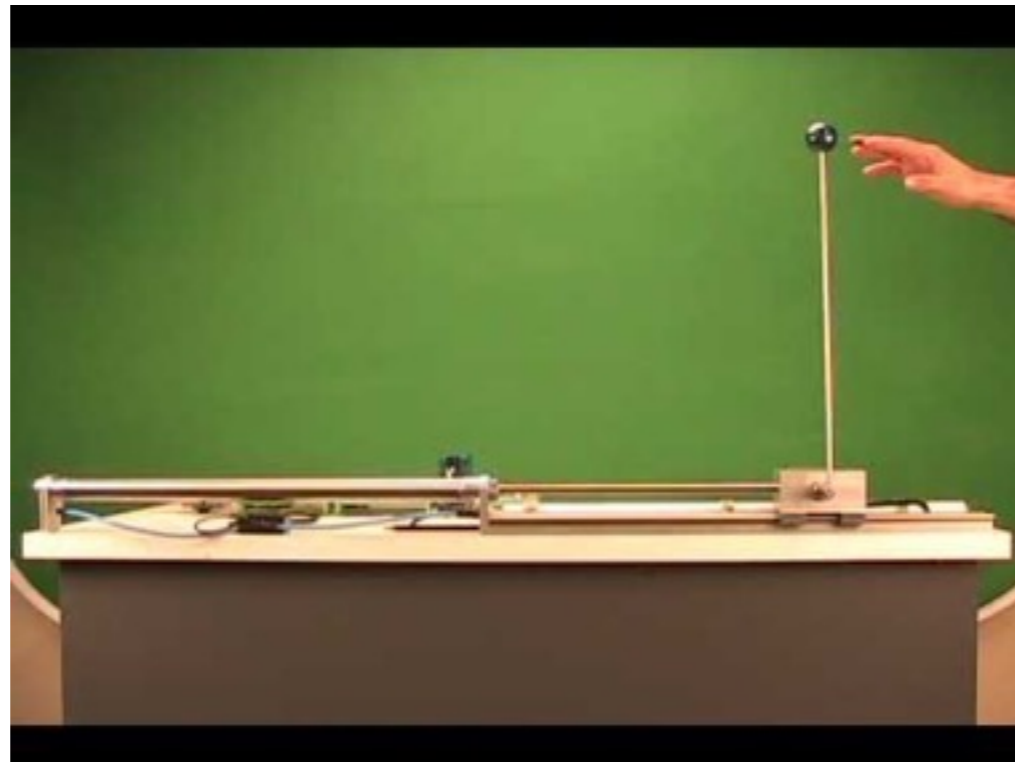 - Worst-case performance or energy

The Seminar

- Reliability

  - Many software systems process input derived from sensors

    - Embedded software

  - Sensor input

The Seminar

- Robustness

  - Stability of control software

  - Operate on non-deterministic environment

The Seminar

- Side channel attacks

  - Break implementation by sensing timing, power etc.

  - Discover private information e.g. secret keys of ATM card

Choosing Papers

- Send an email to sudiptac@st.cs.uni-saarland.de

  - **6 preferences, preference by area and not by paper**

  - **One paper (or "any" paper) from each area, ordered by your preference**

    - Profiling

    - Performance analysis and testing

    - Energy consumption analysis and testing

    - Reliability analysis (only one paper)

    - Robustness analysis and testing

    - Side channels

  - **Deadline: 4th Nov, 2015, 23:59:59**

Assigning Papers

- First-come-first-served

- Order of preference

- It is possible that all your preferences have been assigned

  - I'll contact you to resolve conflict, most likely by suggesting a paper in the area of your preference

Upcoming meetings

- *13th November, 2015:*

  - Efficient Path Profiling (MICRO 1996)

- *20th November, 2015:*

  - Where is the energy spent inside my app?: fine grained energy accounting on smartphones with Eprof (EUROSYS 2012)

- **The paper for next meeting should be assigned now**

Objective

- Learn about non-functional software properties

  - Timing

  - Energy

  - Robustness

  - Security (primarily side-channel leaks)

  - Reliability

The Seminar

- Performance/energy bugs

  - Waste performance or energy

    - Unnecessary memory access

    - Unnecessary GPS access

  - Worst-case performance or energy

Upcoming meetings

- *13th November, 2015:*

  - Efficient Path Profiling (MICRO 1996)

- *20th November, 2015:*

  - Where is the energy spent inside my app?: fine grained energy accounting on smartphones with Eprof (EUROSYS 2012)

- **The paper for next meeting should be assigned now**

Choosing Papers

- Send an email to sudiptac@st.cs.uni-saarland.de

  - **6 preferences, preference by area and not by paper**

  - **One paper (or "any" paper) from each area, ordered by your preference**

    - Profiling

    - Performance analysis and testing

    - Energy consumption analysis and testing

    - Reliability analysis (only one paper)

    - Robustness analysis and testing

    - Side channels

  - **Deadline: 4th Nov, 2015, 23:59:59**