SFM-11: CONNECT, Bertinoro, June 2011

The multi-facets of building dependable applications over connected physical objects

S.C. Cheung
Director of RFID Center
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
http://www.cse.ust.hk/~scc

Due to my own research background, I will address the topic from the perspective of software engineering, and draw examples from RFID applications for illustration.

Yet, the concepts and issues discussed should be generalizable to the physical objects connected using other wireless technologies, such as wifi, zigbee, bluetooth and so on.

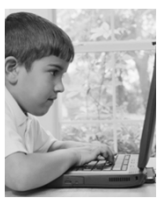
Many interesting issues arise from connected physical objects (CPO). I may not have answers to most of the issues. **BUT**, I will share with you our past experience.

A story from Mr. Bean …

Jerry Bean's diary

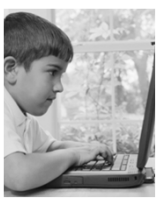
Who is Jerry Bean?

- Jerry is completing his first year of PhD study.
- He understands that PhD study requires a clear mission.
  - Buy house and get married soon after graduation.
- He decides to work on a topic with promising research and business opportunities.
- He has an adviser working on software engineering.



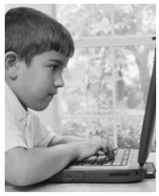
Who is Jerry Bean (cont.)?

Most importantly, his adviser wants him to publish at ICSE, ESEC-FSE, FASE, ISSTA, …



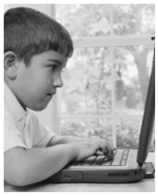
What are connected physical objects (CPO)?

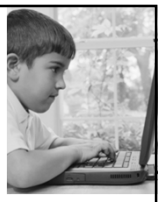
Jerry's story (Part I)



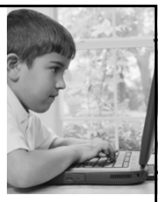
What are connected physical objects?

- Jerry loves YouTube.
- One day, he found a video.

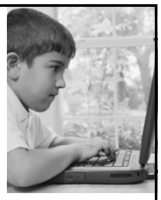




Jerry gets interested and looks further into RFID …



INTERESTING!!
Connected physical objects are shaping our behaviors in every walk of life.



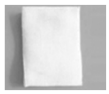
They also help us to meet new standards and guidelines

WHO Guidelines for Safe Surgery 2009

- Objective 7: The team will prevent inadvertent retention of instruments or sponges in surgical wounds.
  - Need to systematically track/count the sponges used in an operation.

RFID-enabled sponges

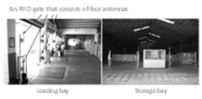
- Real-time tracking and counting sponges used in an operation



RFID-enabled sponge

Connected physical objects (CPO)

- Stay connected.
  - Multiple networking technologies: Wifi, Zigbee, Bluetooth, RFID, …



- Identify themselves actively.
- Feature tight combination of computational and physical elements.
- Tight integration of computation and physical processes.

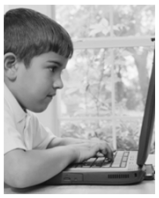
Connected physical objects (CPO)

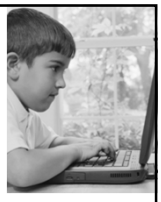
Call for interdisciplinary efforts:

- Algorithm
- Database
- Pervasive computing
- Logistics management
- …
- **Software engineering**

Where are the software engineering research problems?

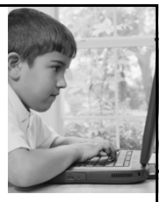
Jerry's story (Part II)





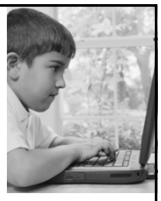
Question

Are there any fundamental differences between CPO systems and conventional software systems?



Question

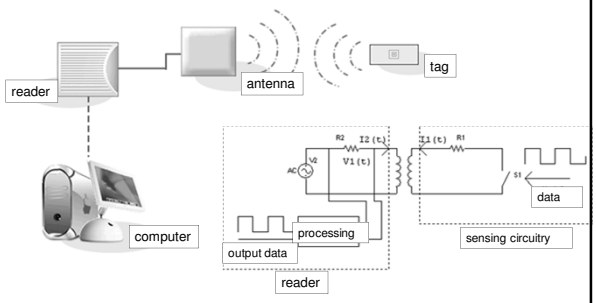
In which way CPO systems induce new challenges?



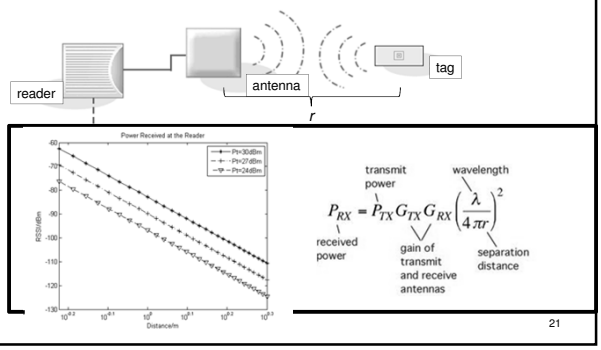
Observation

By being physical, CPOs are subject to physical laws and able to sense/interact with their physical environment

Physical law

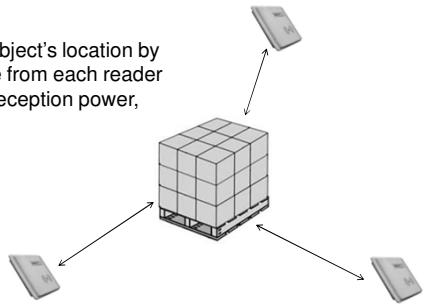


Physical law



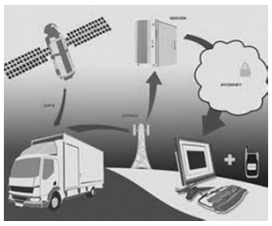
Indoor object tracking

Compute object's location by its distance from each reader based on reception power, e.g., RSSI.



Outdoor object tracking

Driver carries a smart phone



Sensing environment

accelerometer

About 35 percent of people over age 65 fall in their homes at least once each year. That figure increases to 50 percent for those with age 75 and over. -- Havard Health Letters 2009

Adaptive actions

accelerometer

- Raise an alarm
- Call her relatives
- Call her neighbors
- Call emergency services
- Contact her nearest hospital
- Call her family doctor
- Record the frequency

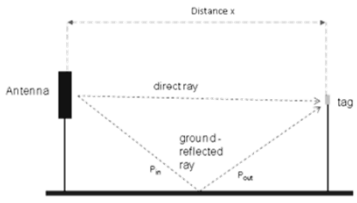
Adaptive actions

accelerometer

- Raise an alarm
- Call her relatives
- Call her neighbors
- Call emergency services
- Contact her nearest hospital
- Call her family doctor
- Record the frequency

Subject to physical laws

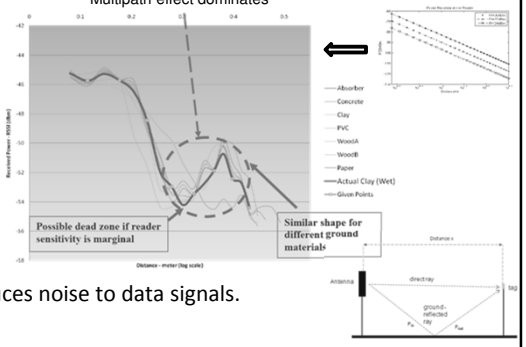
- Physical laws may induce undesirable effects.



Multi-path Effect

Subject to physical laws

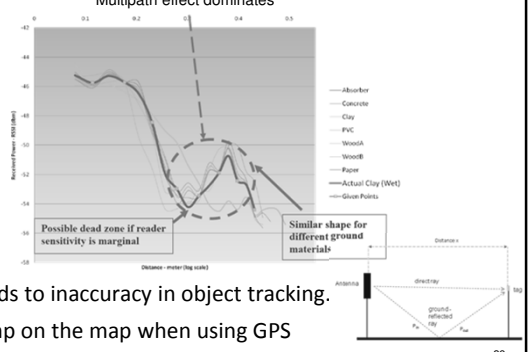
Multipath effect dominates



- Induces noise to data signals.

Subject to physical laws

Multipath effect dominates



- Leads to inaccuracy in object tracking.
- Jump on the map when using GPS

Induces engineering problem

Induces engineering problem

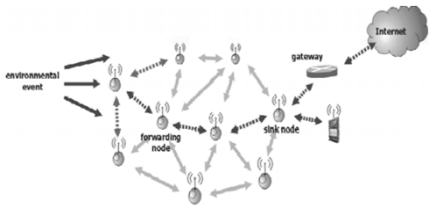
Is the problem common?

- For RFID data, the observed read rate (i.e., the percentage of tags in a reader's vicinity that are actually reported) in real-life deployment may drop down to the range of 60-70% [1]

[1] Jeffery, S.R., Garofalakis, M., and Frankin, M.J. Adaptive Cleaning for RFID Data Streams. In *Proceedings of the 32nd International Conference on Very Large Data Bases*, Seoul, Korea, September 2006, 163-174.

When objects are connected …

- Errors can be propagated from one to another.
- Sensitive information of an object can be derived from partial information held by its peers.



When objects are connected …

ON EVENT bird-detect (loc):

SELECT AVG (light), AVG (temp), event.loc
FROM sensors AS s
WHERE dist (s.loc, event.loc) < 10m
SAMPLE INTERVAL 2s FOR 30s

- Can we apply existing techniques on database application testing?

A nesC program for a WSN mote

```
#1: module OscilloscopeM {
#2:   provides interface StdControl;
#3:   uses {
#4:     interface Leds;
#5:     interface ADC as PhotoSensor;
#6:     interface ADC as TempSensor;
#7:     interface SendMsg; /* other interfaces */
#8:   }
#9: } implementation {
#10:   uint8_t packetReadingNumber;          // shared variable
#11:   // uint8_t lock = 0;
#12:   TOS_Msg msg;                          // shared variable
#13:   task void task4() { /* do something */ }
#14:   task void task3() { /* do something */ }
#15:   task void task2() {
#16:     OscopeMsg *pack;
#17:     static uint32_t i;
#18:     uint32_t start = i;
#19:     atomic {                            // atomic block
#20:       pack = (OscopeMsg *)msg.data;
#21:       packetReadingNumber = 0;
#22:       for (; i < start + 100000 && i < 200001; i++) { /* process sensing data */
#23:       }
#24:       if (i < 200001) {
#25:         post task2(); // post another instance to process remaining data
#26:         return;
#27:       } // else (finish processing the data): send the packet
#28:       i = 0;
#29:       pack->channel = 1;
#30:       pack->sourceMoteID = TOS_LOCAL_ADDRESS;
#31:       if (call SendMsg.send(TOS_UART_ADDR, sizeof(OscopeMsg), &msg)) {
#32:         call Leds.yellowToggle();
#33:       }
#34:     }
#35:     task void task1() { /* do something */
#36:       post task3();
#37:       post task4();
#38:     }
#39:     async event result_t PhotoSensor.dataReady(uint16_t data) {
#40:       OscopeMsg *pack;
#41:       // atomic { if (lock == 1) return; }
#42:       pack = (OscopeMsg *)msg.data;
#43:       atomic {
#44:         pack->data[packetReadingNumber++] = data;
#45:       }
#46:       if (packetReadingNumber == BUFFER_SIZE) {
#47:         // lock = 1;
#48:         post task2();
#49:       }
#50:       return SUCCESS;
#51:     }
#52:     event result_t SendMsg.sendDone(TOS_MsgPtr sent, result_t success) {
#53:       // atomic lock = 0; /* notify the availability of the message buffer */
#54:       return SUCCESS;
#55:     }
#56:     async event result_t TempSensor.dataReady(uint16_t data) { // do something
#57:       post task1();
#58:       return SUCCESS;
#59:     } /* other commands, events, and tasks */
#60: }
```

- How to learn the behavior of a mote?

Summary of challenges

- Data level
  - Ubiquitous sensing noises are not under full control.
  - Natural constraints on physical objects cannot be modeled completely and precisely.
- Logic level
  - Hardly able to consider adequately the real-life complexity of physical objects that are evolvable and noisy.
  - Unanticipated exceptions occur.

Summary of challenges

- Data level
  - Ubiquitous sensing noises are not under full control.
  - Natural constraints on physical objects cannot be modeled completely and precisely.
- Logic level
  - Need to consider adequately the real-life complexity of physical objects that are evolvable and noisy.
  - Need to prepare for unanticipated exceptions.

37

Summary of challenges

- Data level
  - Ubiquitous sensing noises are not under full control.
  - Natural constraints on physical objects cannot be modeled completely and precisely.

➔ Sensing defects

- Logic level
  - Need to consider adequately the real-life complexity of physical objects that are evolvable and noisy.
  - Need to prepare for unanticipated exceptions occur.

38

Summary of challenges

- Data level
  - Ubiquitous sensing noises are not under full control.
  - Natural constraints on physical objects cannot be modeled completely and precisely.

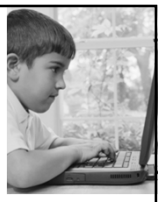
➔ Sensing defects

- Logic level
  - Need to consider adequately the real-life complexity of physical objects that are evolvable and noisy.

➔ Logic defects

  - Need to prepare for unanticipated exceptions occur.

39



Jerry performs more search to check if these defects occur at other CPO systems.

Sensing defects

- Tasker is a popular Android app that performs user-defined tasks based on its environment changes (or situations).

An early bug (in v.20b):

- Failed to detect that wifi was inactive.
- Kept trying to establish connection.
- Drained battery power quickly.

41

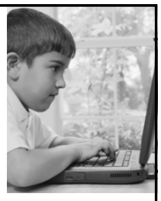
Adaptation defects

- Locale is an award-winning Android app that changes phone's settings based on pre-defined conditions.

An early bug (in v1.0):

- Mistake in condition evaluation.
- May bounced rapidly between adapted settings and default settings occasionally.
- Drained most of the phone processing power.
- Failed to respond to user inputs.

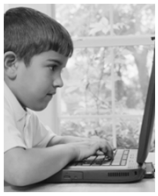
42



After collecting more bug reports, Jerry is interested in defect detection and management.

Defect detection and management

Jerry's story (Part III)



Defect detection

- In most cases, we don't have the ground truth.
- We are not able to tell if a sensed datum (or context) is correct.
  - The meaning of correct is unclear.
  - We need to live with some errors.
- CPO are subject to physical rules.
- Extract constraints that cover context changes from these rules.
- Catch context inconsistency upon rule violation.

Defect detection

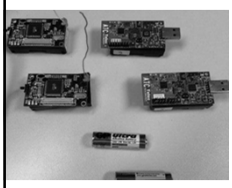
- Data level
  - Ubiquitous sensing noises are not under full control.
  - Natural constraints of physical objects cannot be modeled completely and precisely.

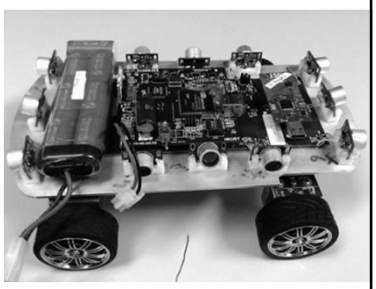
➔ Sensing defects

- Logic level
  - Need to consider adequately the real-life complexity of physical objects that are available for sensing.
  - Need to prepare for unanticipated exceptions occur.

➔ Logic defects

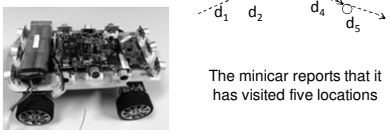
Illustrative Example





Illustrative Example

It is hard to tell which of the five contexts is/are incorrect unless we know the ground truth.

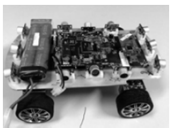


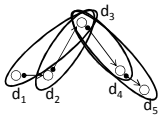
The minicar reports that it has visited five locations

Picture from http://www.thegreenhead.com/2007/09/motorized-walking-star-wars-lego-at-at.php

Illustrative Example

Constraint: The walking speed estimated from location changes should be less than 150% · v





The average speed obtained from the five contexts

Context changes

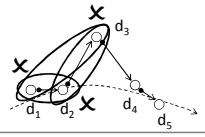
(d_1, d_3), (d_2, d_3), (d_3, d_4), (d_3, d_5)

violate the constraint

Four context inconsistencies found

Illustrative example

Drop-all resolution

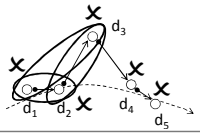


The overcautious nature makes this strategy tend to discard more contexts than necessary

d_3 is discarded, but d_1 and d_2 are discarded

Illustrative example

Drop-all resolution



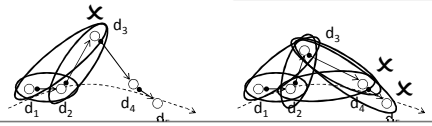
Lesson learnt: We should not discard all inconsistency-related contexts

All data are discarded

Illustrative example

Drop-latest resolution

Move d_3 closer to d_2



Deciding a context immediately to be correct / incorrect based on the nonexistence / existence of any context inconsistency may not give desirable results.

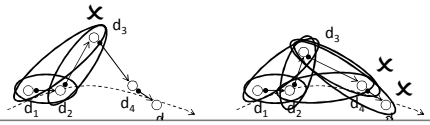
~~Context inconsistency (d_2, d_4)~~

~~Context inconsistency (d_3, d_5)~~

d_3 is kept, while d_4, d_5 are discarded

Illustrative example

Drop-latest resolution



Lesson learnt: We should not make decision based on a single inconsistency-related context.

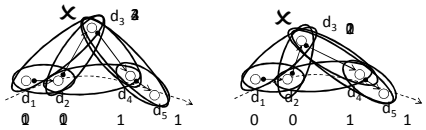
~~Context inconsistency (d_3, d_4)~~

~~Context inconsistency (d_3, d_5)~~

d_3 is kept, while d_4, d_5 are discarded

A possible resolution strategy

Heuristics: A context that participates more frequently in context inconsistencies is more likely to be incorrect



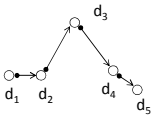
Context inconsistency (d_2, d_3)
Context inconsistency (d_1, d_3)
Context inconsistency (d_3, d_4)
Context inconsistency (d_3, d_5)

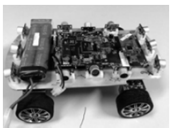
Context inconsistency (d_3, d_4)
Context inconsistency (d_3, d_5)

Illustrative example

Constraint: The walking speed estimated from location changes should be less than 150% · v

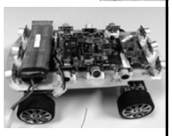
The average speed obtained from the five contexts





Useful constraints should be those that can affect the adaptive behavior of connected physical objects

Research questions



- How to identify useful constraints?
- How to formulate constraints?
- How to model contexts?
- Do we have a time window on the contexts?
- How to evaluate constraints in view of new contexts?
- May multiple context changes be evaluated in different order?
- Can the evaluation be incrementally done?
- What if constraints are changed dynamically?
- May context inconsistencies be automatically resolved?

[Xu et al. ESEC/FSE 2005, ICSE 2006, SEAMS 2007, ICDCS 2008, TOSEM 2010]

Defect detection

- Data level
  - Ubiquitous sensing noises are not under full control.
  - Natural constraints of physical objects cannot be modeled completely and precisely.

➔ Sensing defects

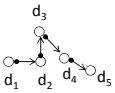
- Logic level
  - Need to consider adequately the real-life complexity of physical objects that are involved in the system.
  - Need to prepare for unanticipated exceptions occur.

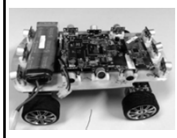
➔ Logic defects

What if logic defects?

Likely induce transient bugs

Hard to catch without a test oracle

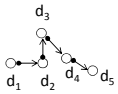




The minicar reports that it has visited five locations

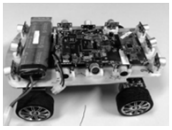
Learning-based defect detection

- Partition context streams into intervals
- Cluster these intervals based on their physical and logical features
- Intervals that cannot be clustered likely contain transient bugs



$d_1d_2d_3d_4d_5d_6$…….d_nd_{n+1}

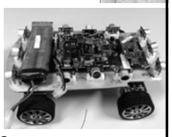
$|d_1d_2d_3d_4d_5|$ $|d_6$…$d_{12}|$ … $|$…$d_nd_{n+1}|$



The minicar reports that it has visited five locations

[Ngai et al. Ad Hoc Networks 2010; Tretmans SFM'11 2011; Zhou et al. ICDCS 2010]

Fault detection of adaptation logic



- How to model adaptation logic at design stage?
- Can we detect adaptation logic at design stage?
- What are adaptation design faults?
- What are the causes of adaptation design faults?
- Do these faults have patterns?
- Can we statically check adaptation design faults?
- Can we generate test cases for CPO systems from their design?
- Do we have new test coverage criteria for CPO systems?

[Lai et al. FSE 2008; Lu et al. FSE 2006; Sama et al. TSE 2010, JSS 2010; Wang et al. ICSE 2007]

Conclusion

CPO systems are written to:

- Bridge the gap between physical objects in daily lives and virtual objects in computers
  - No longer ***static data***, but ***dynamic objects*** that respect and address the evolvable, noisy reality
- Increase the participation and mobility of physical objects in pervasive computation
  - No longer ***passive data***, but ***active processes*** that interact with each other

More research challenges

[Ma, Baresi, Ghezzi et al. ESEC/FSE 2011]

- How to update adaptive components dynamically?
- May dynamical updates be made in peer-to-peer?
- How to model such requirements?
- What is the role of middleware?
- What is a test case?
- How to define test oracles in a CPO system?
- How to prevent failures from propagating among CPOs?
- How to learn the behavior of CPO systems?
- How to enforce security and privacy on CPO?

Dynamic reconfiguration and update

A video prepared by P.R. Kumar
Dept. of Electrical and Computer Engineering, and Coordinated Science Lab
University of Illinois, Urbana-Champaign

video

More research challenges

- How to update adaptive components dynamically?
- May dynamical updates be made in peer-to-peer?
- How to model such requirements?
- What is the role of middleware?
- What is a test case?
- How to define test oracles in a CPO system?
- How to prevent failures from propagating among CPOs?
- How to learn the behavior of CPO systems?
- How to enforce security and privacy on CPO?

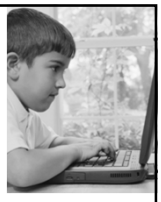
The list is by no means exhaustive

More research challenges

- How to update adaptive components dynamically?
- May dynamical updates be made in peer-to-peer?
- How to model such requirements?
- What is the role of middleware?
- What is a test case?
- How to define test oracles in a CPO system?
- How to prevent failures from propagating among CPOs?
- How to learn the behavior of CPO systems?
- How to enforce security and privacy on CPO?

Solutions to these challenges can involve knowledge beyond software engineering and even computer science

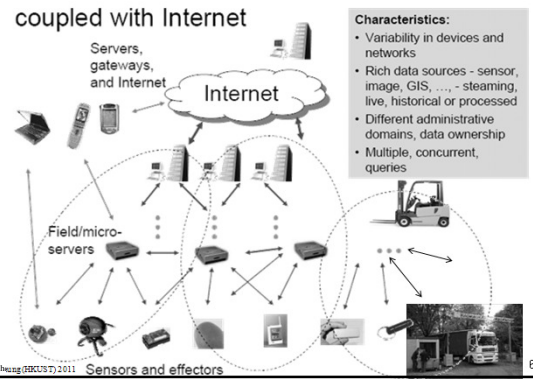
The Internet of Things



A video from IBM
[http://www.youtube.com/watch?v=sfEbMV295Kk]

video

Architecture in-the-large



Further readings

1. L. Capra, W. Emmerich, and C. Mascolo, CARISMA: context-aware reflective middleware system for mobile applications, *IEEE Trans. on Software Engineering 29 (10)*, pp. 929-945, Oct 2003.
2. Betty Cheng and et al., Software Engineering for Self-Adaptive Systems: A Research Roadmap, Software Engineering for Self-Adaptive Systems (2009), *LNCS 5525*, pp. 1-26, Spring 2009.
3. J. Coutaz, J.L. Crowley, S. Dobson, and D. Garlan, Context is key, *Communications of the ACM 48 (3)*, pp. 49-53, Mar 2005.
4. W.G. Griswold, R. Boyer, S.W. Brown, and M.T. Tan, A component architecture for an extensible, highly integrated context-aware computing infrastructure, In *Proc. the 25th International Conf. on Software Engineering*, Portland, USA, pp. 363-372, May 2003.
5. K. Henricksen and J. Indulska, A software engineering framework for context-aware pervasive computing, In *Proc. the 2nd IEEE Conf. on Pervasive Computing and Communications*, Orlando, Florida, USA, pp. 77-86, Mar 2004.
6. Y. Huang, X. Ma, J. Cao, X. Tao, J. Lu, Concurrent event detection for asynchronous consistency checking of pervasive context, In *Proc. the 7th Annual IEEE International Conf. on Pervasive Computing and Communications*, Galveston, Texas, USA, pp. 131-139, Mar 2009.
7. S.R. Jeffery, M. Garofalakis, and M.J. Frankin, Adaptive cleaning for RFID data streams, In *Proc. the 32nd International Conf. on Very Large Data Bases*, Seoul, Korea, pp. 163-174, Sep 2006.
8. C. Julien and G.C. Roman, EgoSpaces: facilitating rapid development of context-aware mobile applications, *IEEE Trans. on Software Engineering 32 (5)*, pp. 281-298, May 2006.

Further readings

9. D. Kulkarni and A. Tripathi, A framework for programming robust context-aware applications, *IEEE Trans. on Software Engineering 36 (2)*, pp. 184-197, Mar/Apr 2010.
10. Zhifeng Lai, S.C. Cheung, W.K. Chan, Inter-Context Control-Flow and Data-Flow Test Adequacy Criteria for nesC Applications, in *Proceedings of the 16th ACM SIGSOFT Symposium on Foundations of Software Engineering (FSE 2008)*, Atlanta, USA, November 9-14, 2008, pp. 94-104.
11. H. Lu, W. Chan, and T. Tse, Testing context-aware middleware-centric programs: a data flow approach and an RFID-based experimentation, In *Proc. the 14th ACM SIGSOFT Symposium on the Foundations of Software Engineering*, Portland, Oregon, USA, pp. 242-252, Nov 2006.
12. Xiaoxing Ma, Luciano Baresi, Carlo Ghezzi, Valerio Panzica La Manna, and Jian Lu, The Onion Patch: Migration in Open Source Ecosystems, in the *Proceedings of ESEC/FSE 2011*, Szeged, Hungary, Sept, 2011.
13. Edith C. H. Ngai, Yangfan Zhou, Michael R. Lyu, Jiangchuan Liu, A delay-aware reliable event reporting framework for wireless sensor-actuator networks. *Ad Hoc Networks 8(7)*: 694-707 (2010).
14. V. Rajamani and C. Julien, Blurring snapshots: temporal inference of missing and uncertain Data, In *Proc. the 8th Annual IEEE International Conf. on Pervasive Computing and Communications*, Mannheim, Germany, pp. 40-50, Mar-Apr 2010.
15. J. Rao, S. Doraiswamy, H. Thakkar, and L.S. Colby, A deferred cleansing method for RFID data analytics, In *Proc. the 32nd International Conf. on Very Large Data Bases*, Seoul, Korea, pp. 175-186, Sep 2006.

Further readings

16. M. Sama, S. Elbaum, F. Raimondi, D.S. Rosenblum, Z. Wang, Context-aware adaptive applications: fault patterns and their automated identification. *IEEE Transactions on Software Engineering 36, 5 (Sep/Oct) 2010*, 644-661.
17. M. Sama, D.S. Rosenblum, Z. Wang, and S. Elbaum, Multi-layer faults in the architectures of mobile, context-aware adaptive applications. *The Journal of Systems and Software, 83 (2010)*, 906-914.
18. Jan Tretmans, Model-Based Testing and Some Steps towards Test-Based Modelling. *SFM 2011, LNCS 6659* (2011), pp. 297-326.
19. Z.L. Wu, C.H. Li, J.K.Y. Ng, and K.R.P.H. Leung, Location estimation via support vector regression, *IEEE Trans. on Mobile Computing 6 (3)*, pp. 311-321, Mar 2007.
20. Chang Xu and S.C. Cheung, Inconsistency Detection and Resolution for Context-Aware Middleware Support, in *Proceedings of the Joint 10th European Software Engineering Conference and 13th ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE 2005)*, Lisbon, Portugal, September 5-9, 2005, pp. 336-345.
21. Chang Xu, S.C. Cheung and W.K. Chan, Incremental Consistency Checking Pervasive Context, Research Paper, in Proceedings of the *28th International Conference on Software Engineering (ICSE 2006)*, Shanghai, China, May 20-28, 2006, pp. 292-301.
22. Chang Xu, S.C. Cheung, W.K. Chan, Goal-Directed Context Validation for Adaptive Ubiquitous Systems, in *Proceedings of the ICSE 2007 Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2007)*, Minneapolis, May 26-27, 2007.

Further readings

24. C. Xu, S.C. Cheung, W.K., Chan, and C. Ye, On impact-oriented automatic resolution of pervasive context inconsistency, In *Proc. the 6th Joint Meeting of the European Software Engineering Conf. and the ACM SIGSOFT Symposium on the Foundations of Software Engineering*, Dubrovnik, Croatia, pp. 569-572, Sep 2007.
25. Chang Xu, S.C. Cheung, W.K. Chan, Chunyang Ye, Heuristics-Based Strategies for Resolving Context Inconsistencies in Pervasive Computing Applications, in *Proceedings of the 28th International Conference on Distributed Computing Systems (ICDCS 2008)*, Beijing, China, June 17-20, 2008, pp. 709-717.
26. Chang Xu, S.C. Cheung, W.K. Chan, and Chunyang Ye, Partial Constraint Checking for Context Consistency in Pervasive Computing, *ACM Transactions on Software Engineering and Methodology*, vol. 19, no. 3, pp. 1-61, January 2010.
27. Yangfan Zhou, Xinyu Chen, M.R. Lyu, and Jiangchuan Liu, Sentomist: Unveiling Transient Sensor Network Bugs via Symptom Mining, *in Proceedings of the 30th International Conference on Distributed Computing Systems (ICDCS 2010)*,Pages 784-794, Genoa, Italy, June 21-25, 2010.
28. Z. Wang, S. Elbaum, and D.S. Rosenblum, Automated generation of context-aware tests. In *Proceedings of the 29th International Conference on Software Engineering*, Minneapolis, MN, USA, May 2007, 406-415.

Some useful journals

- ACM Transactions on Software Engineering and Methodology
- ACM Transactions on Sensor Networks
- ACM Transactions on Embedded Computing Systems
- ACM Transactions on Computer Systems
- ACM Transactions on Programming Languages and Systems
- IEEE Transactions on Software Engineering
- IEEE Transactions on Computers
- IEEE Pervasive Computing
- Real Time Systems
- ACM Transactions on Intelligent Systems and Technology

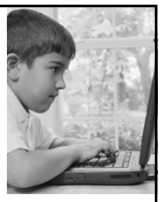
Some useful conferences

- ICSE, ESEC/FSE, ASE, Percom, Ubicom, ICDCS, RTSS, Sensys, EMSOFT, SIGMOD, VLDB, ICDE, FASE, …

Take home messages

- Take advantages of open-source projects
  - Identify real problems
    - Bug reports & discussion forums
  - Collect benchmark subjects
    - Code releases, JUnit tests, test drivers
- Get hands dirty
  - Do a few implementations
  - Get hands-on experience on real programs
  - There are many analysis tools available!!

Thank you



Grazie
Vielen Dank
Merci
Obrigado
Gracias
σας ευχαριστώ

Where is our Jerry?

Jerry's story (Part IV)

He graduated and started his enterpreneurship