2nd International Symposium on High Confidence Software, Qingdao, Oct 2012

Fault Detection for Context-Aware Applications

In collaboration with Chang Xu, Yepang Liu, Xiaoxing Ma, Chun Cao and Jian Lv

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

1. *Yepang Liu, Chang Xu, and S.C. Cheung, "AFChecker: Effective Model Checking for Context-aware Adaptive Applications," submission under review.*
2. *Chang Xu, S.C. Cheung, Xiaoxing Ma, Chun Cao and Jian Lv, "Dynamic Fault Detection in Context-aware Adaptation," Proc. of Internetware 2012, 30-31 October, Qingdao, China.*

Context-Aware Computing

Software

- May not have a fixed behavior
- Continually senses users’ environment
- Adapts its behavior to users need and environment

Context-aware Adaptation

Jenny at Office

Jenny at Home

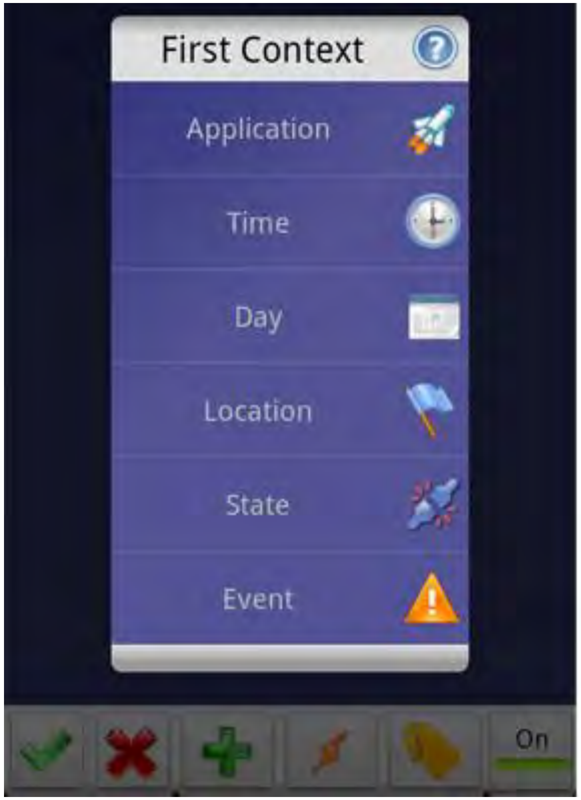
Enabling Technologies

- GPS/Wifi/Bluetooth sensors
- Accelerometer
- Gyroscope
- Proximity sensor
- Digital compass
- Ambient light sensor
- Power sensor
- Microphone
- Camera

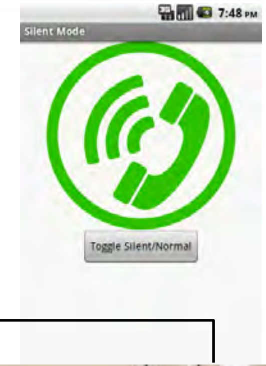
Context-Aware Adaptive Applications

- Increasingly popular
- Locale
- SweetDreams
- Tasker
  - Google contest awardee in 2009

Rule Configuration in Tasker

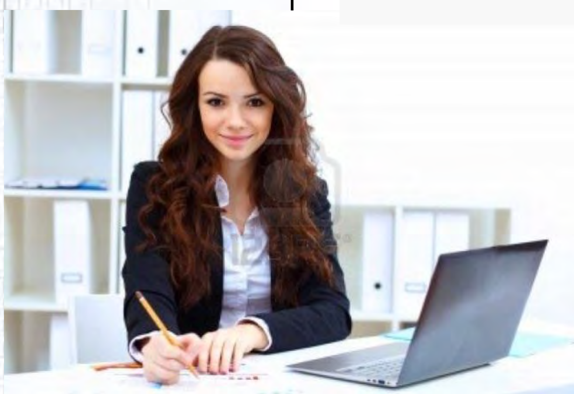


Context-aware Adaptation



- Configured by rules
  - R1: Silent mode when detect a Bluetooth device “OfficePC”
  - R2: Ringing mode when GPS location is at “Home”

Non-deterministic Adaptation



- Configured by rules
  - R1: Silent mode when detect a Bluetooth device “OfficePC”
  - R2: Ringing mode when GPS location is at “Home”

Existing Checking Approaches

- Dynamic approaches
  - Execution on real devices

- Static approaches
  - No real execution
  - Exhaustive state space exploration

Static Rule Misconfiguration Checking

- Extract an adaptation finite state machine

Adaptation Rules of PhoneAdapter [Sama et al., TSE10]

Rule Name	Current States	New State	Full Predicate	Simple Predicate	Priority
ActivateOutdoor	General	Outdoor	GPS.isValid() ∧ ¬GPS.location()=home ∧ ¬GPS.location()=office	$A_{gps} \wedge \neg B_{gps} \wedge \neg C_{gps}$	5
DeactivateOutdoor	Outdoor	General	¬ActivateOutdoor	$\neg(A_{gps} \wedge \neg B_{gps} \wedge \neg C_{gps})$	5
ActivateJogging	Outdoor	Jogging	GPS.isValid() ∧ GPS.speed()> 5	$A_{gps} \wedge D_{gps}$	5
DeactivateJogging	Jogging	Outdoor	¬ActivateJogging	$\neg(A_{gps} \wedge D_{gps})$	5
ActivateDriving	General, Home, Office, Outdoor	Driving	BT=car_handsfree	A_{bt}	1
DeactivateDriving	Driving	General	¬ActivateDriving	$\neg A_{bt}$	1
ActivateDrivingFast	Driving	DrivingFast	GPS.isValid() ∧ GPS.speed()> 70	$A_{gps} \wedge E_{gps}$	0
DeactivateDrivingFast	DrivingFast	Driving	¬ActivateDrivingFast	$\neg(A_{gps} \wedge E_{gps})$	0
ActivateHome	General	Home	BT=home_pc ∨ (GPS.isValid() ∧ GPS.location()=home)	$B_{bt} \vee (A_{gps} \wedge B_{gps})$	5
DeactivateHome	Home	General	¬ActivateHome	$\neg(B_{bt} \vee (A_{gps} \wedge B_{gps}))$	5
ActivateOffice	General	Office	BT=office_pc ∨ BT=office_pc_* ∨ (GPS.isValid() ∧ GPS.location()=office)	$C_{bt} \vee D_{bt} \vee (A_{gps} \wedge C_{gps})$	5
DeactivateOffice	Office	General	¬ActivateOffice	$\neg(C_{bt} \vee D_{bt} \vee (A_{gps} \wedge C_{gps}))$	5
ActivateMeeting	Office	Meeting	Time>=meeting_start ∧ BT.count()>= 3	$A_t \wedge E_{bt}$	4
DeactivateMeeting	Meeting	Office	Time>=meeting_end	B_t	4
ActivateSync	General	Sync	BT=home_pc ∨ BT=office_pc	$B_{bt} \vee C_{bt}$	9
DeactivateSync	Sync	General	¬ActivateSync	$\neg(B_{bt} \vee C_{bt})$	9

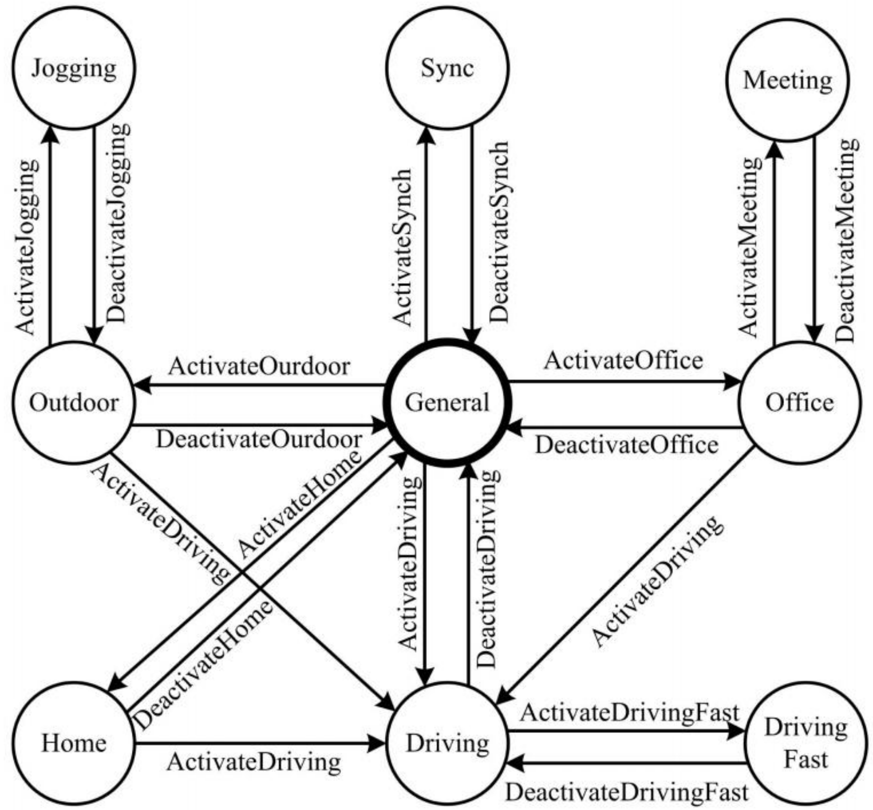
Michele Sama, Sebastian Elbaum, Franco Raimondi, David S. Rosenblum, Zhimin Wang, IEEE TSE 36(5), Sep/Oct 2010.

Static Rule Misconfiguration Checking

- Extract an adaptation finite state machine

Adaptation Rules of PhoneAdapter

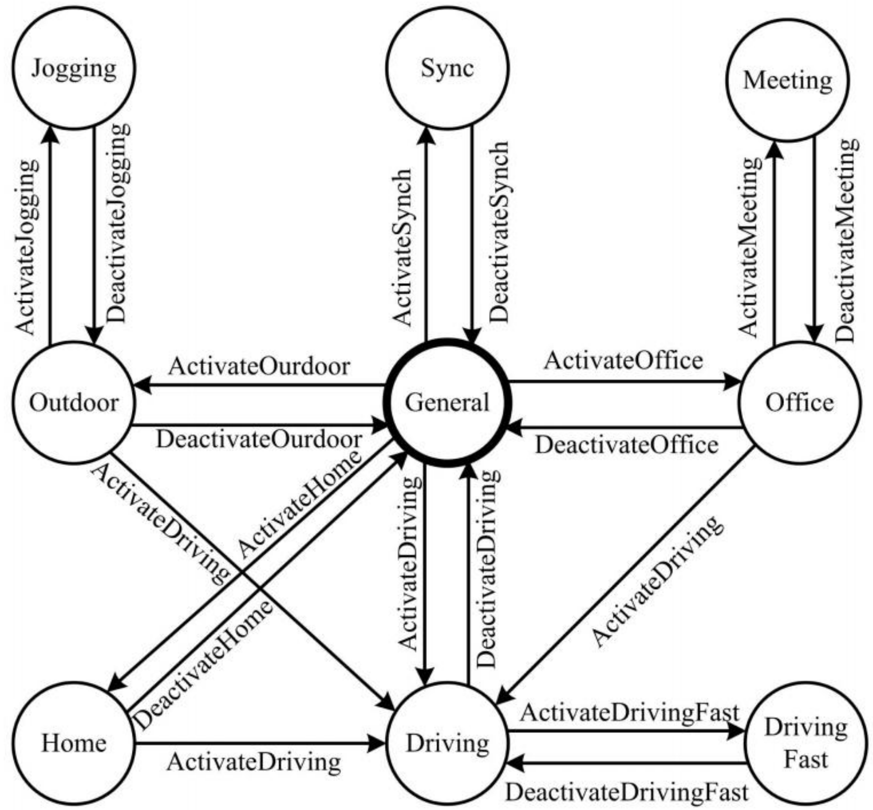
Rule Name	Current States	New State	Full Predicate	Simple Predicate	Priority
ActivateOutdoor	General	Outdoor	GPS.isValid() ∧ ¬GPS.location()=home ∧ ¬GPS.location()=office	$A_{gps} \wedge \neg B_{gps} \wedge \neg C_{gps}$	5
DeactivateOutdoor	Outdoor	General	¬ActivateOutdoor	$\neg(A_{gps} \wedge \neg B_{gps} \wedge \neg C_{gps})$	5
ActivateJogging	Outdoor	Jogging	GPS.isValid() ∧ GPS.speed()> 5	$A_{gps} \wedge D_{gps}$	5
DeactivateJogging	Jogging	Outdoor	¬ActivateJogging	$\neg(A_{gps} \wedge D_{gps})$	5
ActivateDriving	General, Home, Office, Outdoor	Driving	BT=car_handsfree	A_{bt}	1
DeactivateDriving	Driving	General	¬ActivateDriving	$\neg A_{bt}$	1
ActivateDrivingFast	Driving	DrivingFast	GPS.isValid() ∧ GPS.speed()> 70	$A_{gps} \wedge E_{gps}$	0
DeactivateDrivingFast	DrivingFast	Driving	¬ActivateDrivingFast	$\neg(A_{gps} \wedge E_{gps})$	0
ActivateHome	General	Home	BT=home_pc ∨ (GPS.isValid() ∧ GPS.location()=home)	$B_{bt} \vee (A_{gps} \wedge B_{gps})$	5
DeactivateHome	Home	General	¬ActivateHome	$\neg(B_{bt} \vee (A_{gps} \wedge B_{gps}))$	5
ActivateOffice	General	Office	BT=office_pc ∨ BT=office_pc_* ∨ (GPS.isValid() ∧ GPS.location()=office)	$C_{bt} \vee D_{bt} \vee (A_{gps} \wedge C_{gps})$	5
DeactivateOffice	Office	General	¬ActivateOffice	$\neg(C_{bt} \vee D_{bt} \vee (A_{gps} \wedge C_{gps}))$	5
ActivateMeeting	Office	Meeting	Time>=meeting_start ∧ BT.count()>= 3	$A_t \wedge E_{bt}$	4
DeactivateMeeting	Meeting	Office	Time>=meeting_end	B_t	4
ActivateSync	General	Sync	BT=home_pc ∨ BT=office_pc	$B_{bt} \vee C_{bt}$	9
DeactivateSync	Sync	General	¬ActivateSync	$\neg(B_{bt} \vee C_{bt})$	9



[Sama et al., TSE10]

Static Rule Misconfiguration Checking

- Model check finite state machine against given global constraints using OBDDs



1. GPS sensor must be on before checking locations
2. Locations are mutually exclusive
3. Speed increases monotonically
4. Meeting must start before it ends

[Sama et al., TSE10]

Static Rule Misconfiguration Checking

PhoneAdapter with 9 states and 19 rules

State	Nondeterministic Adaptations	Dead Predicates	Adaptation Races	Adaptation Cycles	Unreachable States
General	37	1	45	13	0
Outdoor	3	0	135	23	0
Jogging	0	0	97	19	0
Driving	0	0	36	13	0
DrivingFast	0	0	58	19	0
Home	0	0	76	19	0
Office	0	0	29	1	0
Meeting	0	0	32	1	0
Sync	0	0	27	5	1

Challenges in Static Checking

- Could generate many false positives
  - Real world constraints are not precisely modeled

Adaptation Rules of PhoneAdapter [Sama et al., TSE10]

Rule Name	Current States	New State	Full Predicate	Simple Predicate	Priority
ActivateOutdoor	General	Outdoor	GPS.isValid() ∧ ¬GPS.location()=home ∧ ¬GPS.location()=office	$A_{gps} \wedge \neg B_{gps} \wedge \neg C_{gps}$	5
DeactivateOutdoor	Outdoor	General	¬ActivateOutdoor	$\neg(A_{gps} \wedge \neg B_{gps} \wedge \neg C_{gps})$	5
ActivateJogging	Outdoor	Jogging	GPS.isValid() ∧ GPS.speed()> 5	$A_{gps} \wedge D_{gps}$	5
DeactivateJogging	Jogging	Outdoor	¬ActivateJogging	$\neg(A_{gps} \wedge D_{gps})$	5
ActivateDriving	General, Home, Office, Outdoor	Driving	BT=car_handsfree	A_{bt}	1
DeactivateDriving	Driving	General	¬ActivateDriving	$\neg A_{bt}$	1
ActivateDrivingFast	Driving	DrivingFa	GPS.isValid() ∧ GPS.speed()> 70	$A_{gps} \wedge E_{gps}$	0
DeactivateDrivingFast	DrivingFast	Driving	¬ActivateDrivingFast	$\neg(A_{gps} \wedge E_{gps})$	0
ActivateHome	General	Home	BT=home_pc ∨ (GPS.isValid() ∧ GPS.location()=home)	$B_{bt} \vee (A_{gps} \wedge B_{gps})$	5
DeactivateHome	Home	General	¬ActivateHome	$\neg(B_{bt} \vee (A_{gps} \wedge B_{gps}))$	5
ActivateOffice	General	Office	BT=office_pc ∨ BT=office_pc_* ∨ (GPS.isValid() ∧ GPS.location()=office)	$C_{bt} \vee D_{bt} \vee (A_{gps} \wedge C_{gps})$	5
DeactivateOffice	Office	General	¬ActivateOffice	$\neg(C_{bt} \vee D_{bt} \vee (A_{gps} \wedge C_{gps}))$	5
ActivateMeeting	Office	Meeting	Time>=meeting_start ∧ BT.count()>= 3	$A_t \wedge E_{bt}$	4
DeactivateMeeting	Meeting	Office	Time>=meeting_end	B_t	4
ActivateSync	General	Sync	BT=home_pc ∨ BT=office_pc	$B_{bt} \vee C_{bt}$	9
DeactivateSync	Sync	General	¬ActivateSync	$\neg(B_{bt} \vee C_{bt})$	9

Challenges in Static Checking (1)

- Could generate many false positives
  - Real world constraints are not precisely modeled

Adaptation Rules of PhoneAdapter [Sama et al., TSE10]

Rule Name	Current States	New State	Full Predicate	Simple Predicate	Priority
ActivateOutdoor	General	Outdoor	GPS.isValid() ∧ ¬GPS.location()=home ∧ ¬GPS.location()=office	$A_{gps} \wedge \neg B_{gps} \wedge \neg C_{gps}$	5
DeactivateOutdoor	Outdoor	General	¬ActivateOutdoor	$\neg(A_{gps} \wedge \neg B_{gps} \wedge \neg C_{gps})$	5
ActivateJogging	Outdoor	Jogging	GPS.isValid() ∧ GPS.speed()> 5	$A_{gps} \wedge D_{gps}$	5
DeactivateJogging	Jogging	Outdoor	¬ActivateJogging	$\neg(A_{gps} \wedge D_{gps})$	5
ActivateDriving	General, Home, Office, Outdoor	Driving	BT=car_handsfree	A_{bt}	1
DeactivateDriving	Driving	General	¬ActivateDriving	$\neg A_{bt}$	1
ActivateDrivingFast	Driving	DrivingFast	GPS.isValid() ∧ GPS.speed()> 70	$A_{gps} \wedge E_{gps}$	0
DeactivateDrivingFast	DrivingFast	Driving	¬ActivateDrivingFast	$\neg(A_{gps} \wedge E_{gps})$	0
ActivateHome	General	Home	BT=home_pc ∨ (GPS.isValid() ∧ GPS.location()=home)	$B_{bt} \vee (A_{gps} \wedge B_{gps})$	5
DeactivateHome	Home	General	¬ActivateHome	$\neg(B_{bt} \vee (A_{gps} \wedge B_{gps}))$	5
ActivateOffice	General	Office	BT=office_pc ∨ BT=office_pc_* ∨ (GPS.isValid() ∧ GPS.location()=office)	$C_{bt} \vee D_{bt} \vee (A_{gps} \wedge C_{gps})$	5
DeactivateOffice	Office	General	¬ActivateOffice	$\neg(C_{bt} \vee D_{bt} \vee (A_{gps} \wedge C_{gps}))$	5
ActivateMeeting	Office	Meeting	Time>=meeting_start ∧ BT.count()>= 3	$A_t \wedge E_{bt}$	4
DeactivateMeeting	Meeting	Office	Time>=meeting_end	B_t	4
ActivateSync	General	Sync	BT=home_pc ∨ BT=office_pc	$B_{bt} \vee C_{bt}$	9
DeactivateSync	Sync	General	¬ActivateSync	$\neg(B_{bt} \vee C_{bt})$	9

Challenges in Static Checking (1)

- Could generate many false positives
  - Real world constraints are not precisely modeled

BT=home_pc ∨ (GPS.isValid() ∧ GPS.location()=home)	$B_{bt} \lor (A_{gps} \land \boxed{B_{gps}})$
¬ActivateHome	$\neg(B_{bt} \lor (A_{gps} \land B_{gps}))$
BT=office_pc ∨ BT=office_pc_* ∨ (GPS.isValid() ∧ GPS.location()=office)	$C_{bt} \lor D_{bt} \lor (A_{gps} \land \boxed{C_{gps}})$

  - Model checker based on propositional atoms cannot tell location_at_home (B_{gps}) and location_at_office (C_{gps}) are mutually exclusive.
  - Model checker would explore the state where these two propositional atoms are true at the same time.

Challenges in Static Checking (2)

- Could generate many false positives
  - Real world constraints are not precisely modeled

Adaptation Rules of PhoneAdapter [Sama et al., TSE10]

Rule Name	Current States	New State	Full Predicate	Simple Predicate	Priority
ActivateOutdoor	General	Outdoor	GPS.isValid() ∧ ¬GPS.location()=home ∧ ¬GPS.location()=office	$A_{gps} \wedge \neg B_{gps} \wedge \neg C_{gps}$	5
DeactivateOutdoor	Outdoor	General	¬ActivateOutdoor	$\neg(A_{gps} \wedge \neg B_{gps} \wedge \neg C_{gps})$	5
ActivateJogging	Outdoor	Jogging	GPS.isValid() ∧ GPS.speed()> 5	$A_{gps} \wedge D_{gps}$	5
DeactivateJogging	Jogging	Outdoor	¬ActivateJogging	$\neg(A_{gps} \wedge D_{gps})$	5
ActivateDriving	General, Home, Office, Outdoor	Driving	BT=car_handsfree	A_{bt}	1
DeactivateDriving	Driving	General	¬ActivateDriving	$\neg A_{bt}$	1
ActivateDrivingFast	Driving	DrivingFast	GPS.isValid() ∧ GPS.speed()> 70	$A_{gps} \wedge E_{gps}$	0
DeactivateDrivingFast	DrivingFast	Driving	¬ActivateDrivingFast	$\neg(A_{gps} \wedge E_{gps})$	0
ActivateHome	General	Home	BT=home_pc ∨ (GPS.isValid() ∧ GPS.location()=home)	$B_{bt} \vee (A_{gps} \wedge B_{gps})$	5
DeactivateHome	Home	General	¬ActivateHome	$\neg(B_{bt} \vee (A_{gps} \wedge B_{gps}))$	5
ActivateOffice	General	Office	BT=office_pc ∨ BT=office_pc_* ∨ (GPS.isValid() ∧ GPS.location()=office)	$C_{bt} \vee D_{bt} \vee (A_{gps} \wedge C_{gps})$	5
DeactivateOffice	Office	General	¬ActivateOffice	$\neg(C_{bt} \vee D_{bt} \vee (A_{gps} \wedge C_{gps}))$	5
ActivateMeeting	Office	Meeting	Time>=meeting_start ∧ BT.count()>= 3	$A_t \wedge E_{bt}$	4
DeactivateMeeting	Meeting	Office	Time>=meeting_end	B_t	4
ActivateSync	General	Sync	BT=home_pc ∨ BT=office_pc	$B_{bt} \vee C_{bt}$	9
DeactivateSync	Sync	General	¬ActivateSync	$\neg(B_{bt} \vee C_{bt})$	9

Challenges in Static Checking (2)

- Could generate many false positives
  - Real world constraints are not precisely modeled

Jogging	GPS.isValid() $\wedge$ GPS.speed()> 5	$A_{gps} \wedge D_{gps}$
Outdoor	¬ActivateJogging	$\neg(A_{gps} \wedge D_{gps})$
Driving	BT=car_handsfree	A_{bt}
General	¬ActivateDriving	$\neg A_{bt}$
DrivingFast	GPS.isValid() $\wedge$ GPS.speed()> 70	$A_{gps} \wedge E_{gps}$

  - Model checker cannot tell that a user who is Jogging must not be DrivingFast at the same time.
  - Model checker may report false alarms where Jogging and DrivingFast are simultaneously true.

Challenges in Static Checking (3)

- Could generate many false positives
  - Real world constraints are not precisely modeled

Adaptation Rules of PhoneAdapter [Sama et al., TSE10]

Rule Name	Current States	New State	Full Predicate	Simple Predicate	Priority
ActivateOutdoor	General	Outdoor	GPS.isValid() $\wedge$ $\neg$GPS.location()=home $\wedge$ $\neg$GPS.location()=office	$A_{gps} \wedge \neg B_{gps} \wedge \neg C_{gps}$	5
DeactivateOutdoor	Outdoor	General	$\neg$ActivateOutdoor	$\neg(A_{gps} \wedge \neg B_{gps} \wedge \neg C_{gps})$	5
ActivateJogging	Outdoor	Jogging	GPS.isValid() $\wedge$ GPS.speed()$>$ 5	$A_{gps} \wedge D_{gps}$	5
DeactivateJogging	Jogging	Outdoor	$\neg$ActivateJogging	$\neg(A_{gps} \wedge D_{gps})$	5
ActivateDriving	General, Home, Office, Outdoor	Driving	BT=car_handsfree	A_{bt}	1
DeactivateDriving	Driving	General	$\neg$ActivateDriving	$\neg A_{bt}$	1
ActivateDrivingFast	Driving	DrivingFast	GPS.isValid() $\wedge$ GPS.speed()$>$ 70	$A_{gps} \wedge E_{gps}$	0
DeactivateDrivingFast	DrivingFast	Driving	$\neg$ActivateDrivingFast	$\neg(A_{gps} \wedge E_{gps})$	0
ActivateHome	General	Home	BT=home_pc $\vee$ (GPS.isValid() $\wedge$ GPS.location()=home)	$B_{bt} \vee (A_{gps} \wedge B_{gps})$	5
DeactivateHome	Home	General	$\neg$ActivateHome	$\neg(B_{bt} \vee (A_{gps} \wedge B_{gps}))$	5
ActivateOffice	General	Office	BT=office_pc $\vee$ BT=office_pc_* $\vee$ (GPS.isValid() $\wedge$ GPS.location()=office)	$C_{bt} \vee D_{bt} \vee (A_{gps} \wedge C_{gps})$	5
DeactivateOffice	Office	General	$\neg$ActivateOffice	$\neg(C_{bt} \vee D_{bt} \vee (A_{gps} \wedge C_{gps}))$	5
ActivateMeeting	Office	Meeting	Time$>=$meeting_start $\wedge$ BT.count()$>=$ 3	$A_t \wedge E_{bt}$	4
DeactivateMeeting	Meeting	Office	Time$>=$meeting_end	B_t	4
ActivateSync	General	Sync	BT=home_pc $\vee$ BT=office_pc	$B_{bt} \vee C_{bt}$	9
DeactivateSync	Sync	General	$\neg$ActivateSync	$\neg(B_{bt} \vee C_{bt})$	9

Challenges in Static Checking (3)

- Could generate many false positives
  - Real world constraints are not precisely modeled

BT=office_pc $\vee$ BT=office_pc_* $\vee$ (GPS.isValid() $\wedge$ GPS.location()=office)	$C_{bt} \vee D_{bt} \vee (A_{gps} \wedge C_{gps})$

  - There is no receivable GPS signals in Jenny’s office.
  - GPS reception (A_{gps}) and location_at_office (C_{gps}) virtually exclude each other.
  - Faults reported by model checker occurring under valid GPS reception and location at office are likely spurious.

Challenges in Static Checking

- Could generate many false positives
  - Real world constraints are not precisely modeled
  - Over-approximate what can actually happen

Two important factors determine adaptive behaviors

  - Application internal logic
  - Dynamics of external environment

Challenges in Static Checking

- Could generate many false positives
- Two important factors determine adaptive behaviors
  - **Application internal logic**
  - Dynamics of external environment

slowDriving = *true* if $20 \le GPS.speed < 70$
fastDriving = *true* if $70 \le GPS.speed < 350$

- Use constraint solver to identify logically inconsistent situations
- States corresponding to inconsistent situations are not explored in model checking

Challenges in Static Checking

- Could generate many false positives
- Two important factors determine adaptive behaviors
  - Application internal logic
  - **Dynamics of external environment**

Multiple ways to deduce location context

Bluetooth

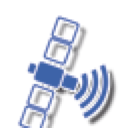
GPS

Challenges in Static Checking

- Could generate many false positives
- Two important factors determine adaptive behaviors
  - Application internal logic
  - **Dynamics of external environment**

Reception of GPS is poor in Jenny's office

Bluetooth



GPS

Challenges in Static Checking

- Could generate many false positives
- Two important factors determine adaptive behaviors
  - Application internal logic
  - **Dynamics of external environment**

Jenny often turns off GPS in office to save battery

Bluetooth

GPS

Modeling the Dynamic Environment

- Dynamic of external environment
  - Cannot always be addressed using internal logic & constraint solvers
  - Varies across users
  - But such implicit knowledge is often pre-assumed by users

Jenny often turns off GPS in office to save battery

Bluetooth

GPS

Modeling the Dynamic Environment

- Can dynamic environment be inferred?
  - Environment (include user habit) is intrinsically complex
  - Infer only those relevant to the configured rules and software logic?
  - Infer only those help suppress spurious fault reports?

Jenny often turns off GPS in office to save battery

Bluetooth

GPS

Modeling the Dynamic Environment

- Jenny sometimes travels with her notebook.
- Jenny often stays at home during weekends.
- Jenny often turns off GPS while she is in office.

Jenny often turns off GPS in office to save battery

Bluetooth

GPS

Modeling the Dynamic Environment

- Environment model inference
  - Sensory data (what, when, how, privacy)
  - Probabilistic correlation (association rule learning, support, confidence)
  - Ranking function of fault reports

Jenny often turns off GPS in office to save battery

Bluetooth

GPS

Modeling the Dynamic Environment

- Environment model validation
  - User (partial validation?)
  - Carefully generated test scenarios
  - Feedback from user’s confirmation of earlier fault reports

Jenny often turns off GPS in office to save battery

Bluetooth

GPS

High Confidence Systems with Big Data

- How to make a software system of high confidence in the World of Big Data?
- Where are the software engineering problems?

High Confidence Systems with Big Data

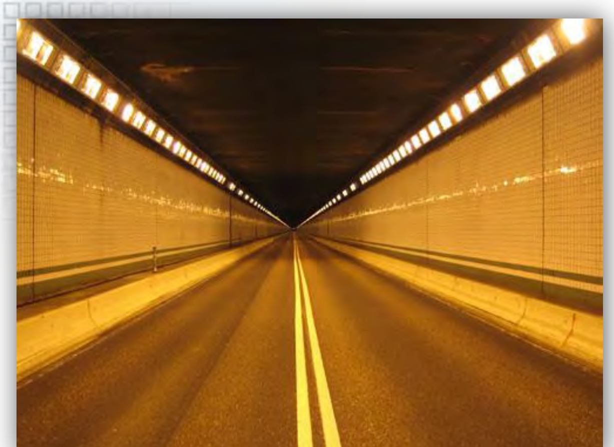
- Big data challenges
  - Data are inherently noisy

No precise oracle to identify noises!

Maybe incomplete (lost in tunnel)

Maybe inaccurate (absorbed, skewed by metal buildings)

May even conflict (interfering)



High Confidence Systems with Big Data

- Big data challenges (cont.)
  - Have to tolerate noisy data

- Detecting and resolving inconsistencies is a way of tolerance, but it has to be
  - efficient (time critical)
  - automated (requires little human intervention)
  - application-specific