High-Confidence Ubiquitous Computing Systems

David S. Rosenblum
School of Computing
National University of Singapore

Some Facts

Google Android Market

- The average price of the top 50 *paid* applications is US$3.79 [modymi.com]
- 79.3% of paid applications have been downloaded *less than 100 times* [Distimo]
- Only 0.1% of paid applications have been downloaded 50,000 times or more [Distimo]

Some Facts

Google Android Market

- The average price of the top 50 *paid* applications is US$3.79 [modymi.com]
- 79.3% of paid applications have been downloaded *less than 100 times* [Distimo]
- Only 0.1% of paid applications have been downloaded 50,000 times or more [Distimo]

There are many simplistic, low-quality apps!

CAAAs

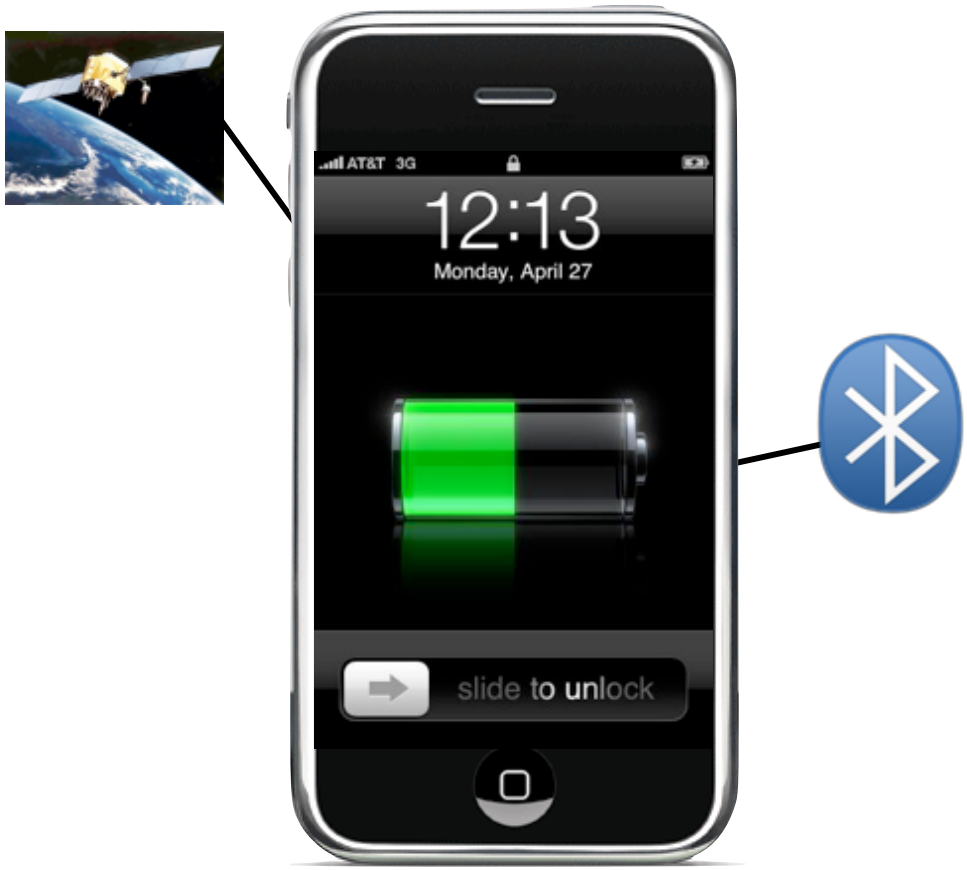
Context-Aware Adaptive Applications

CAAAs

Context-Aware Adaptive Applications

CAAAs

Context-Aware Adaptive Applications

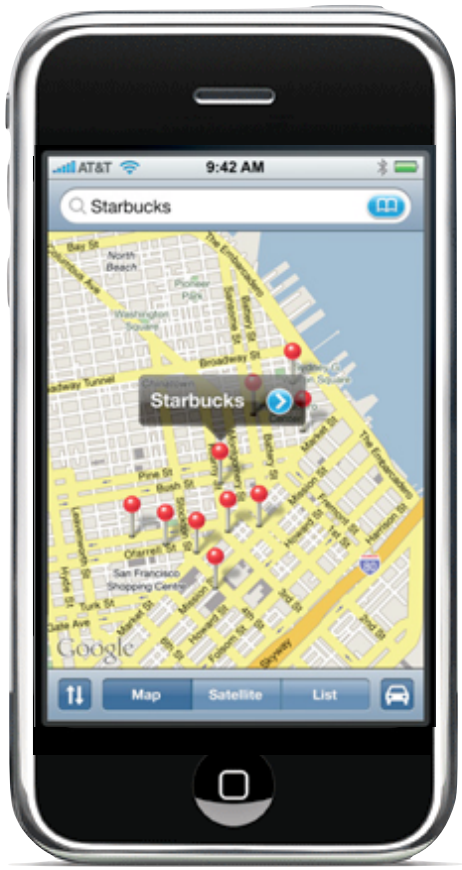


CAAAs

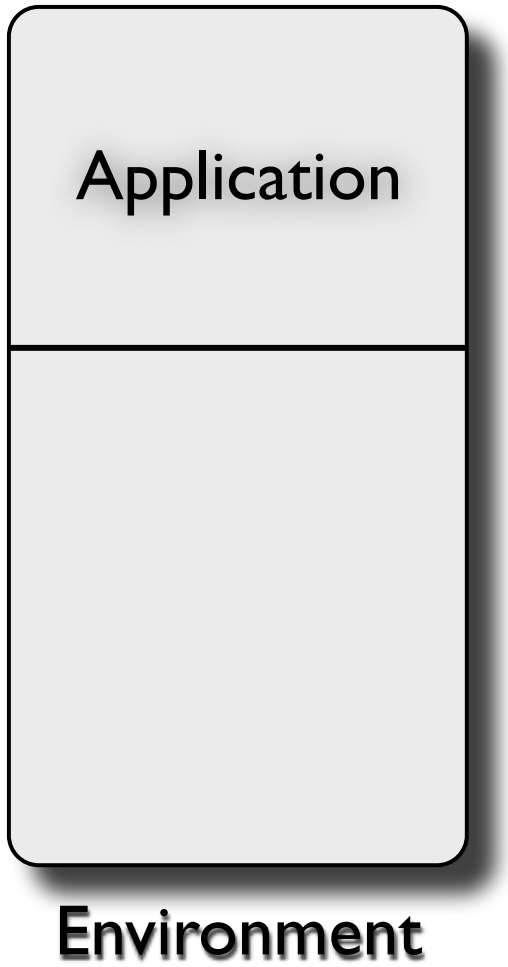
Context-Aware **Adaptive** Applications

CAAAs

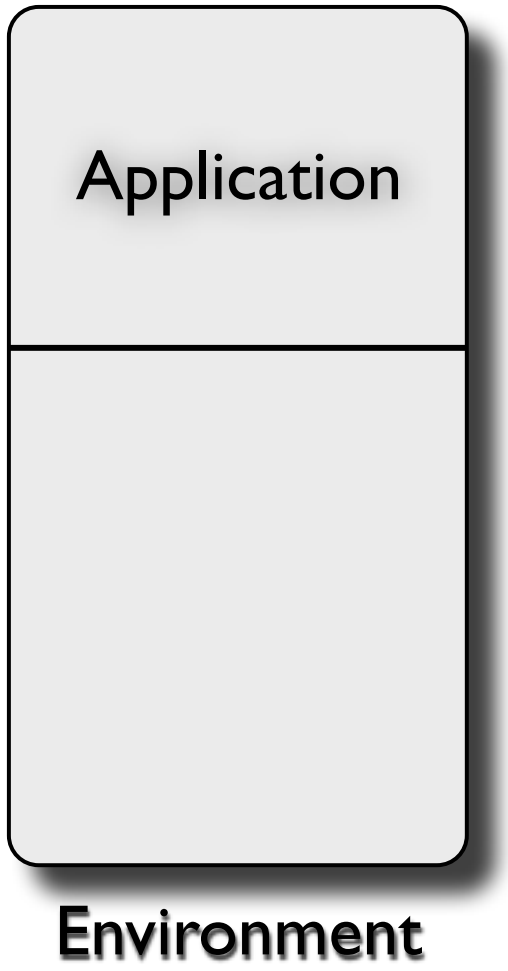
Context-Aware **Adaptive** Applications



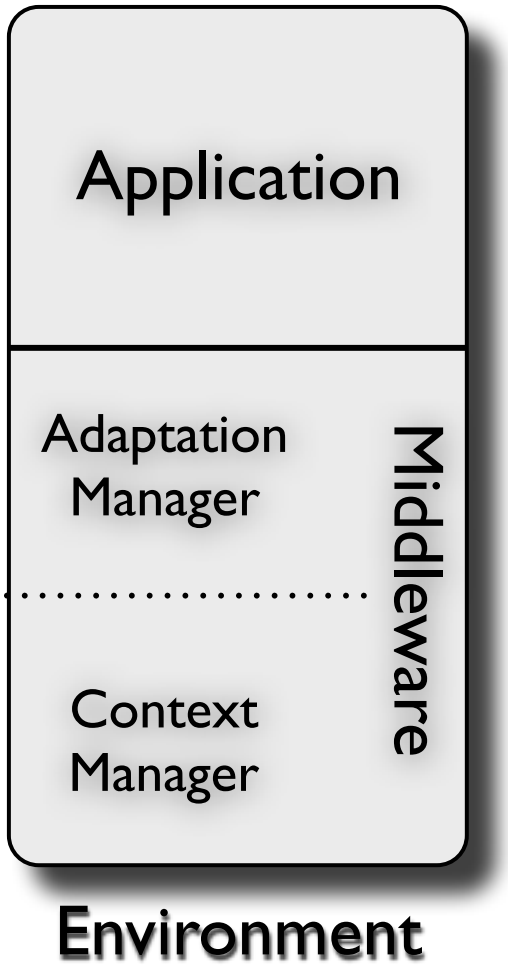
Adaptation in CAAAs



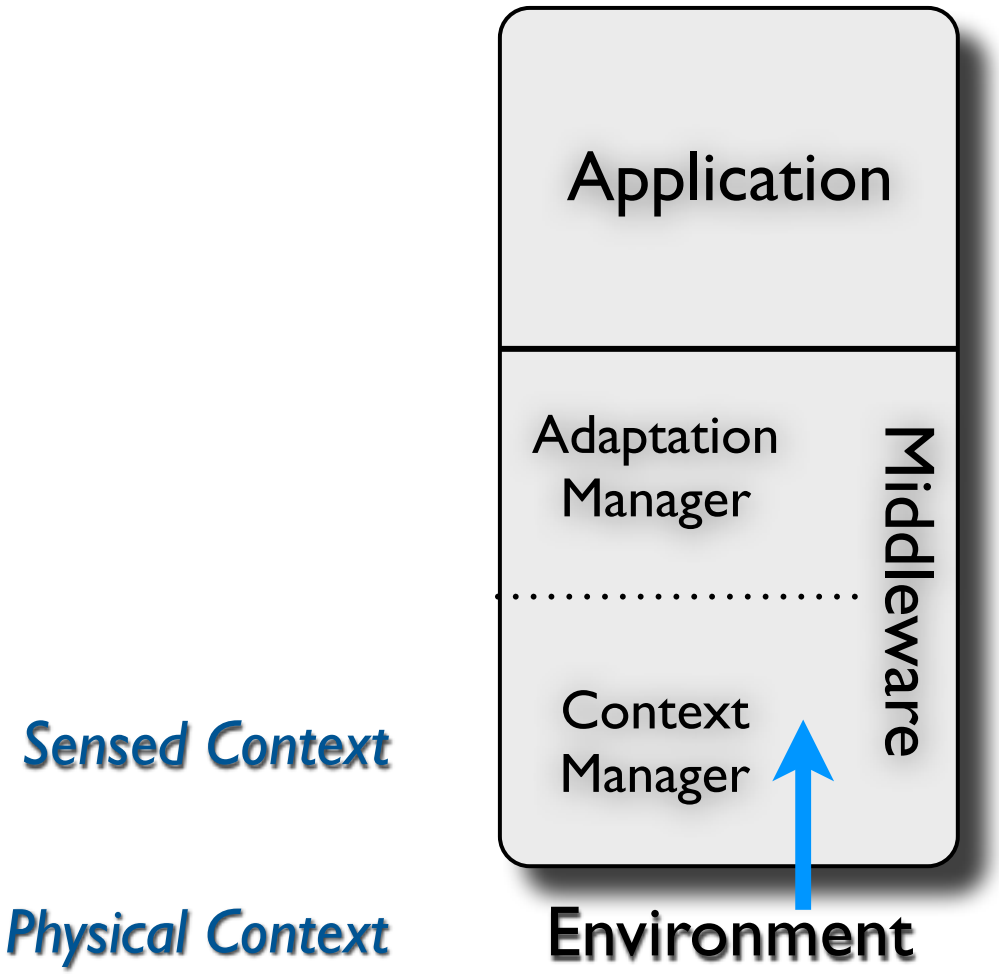
Adaptation in CAAAs



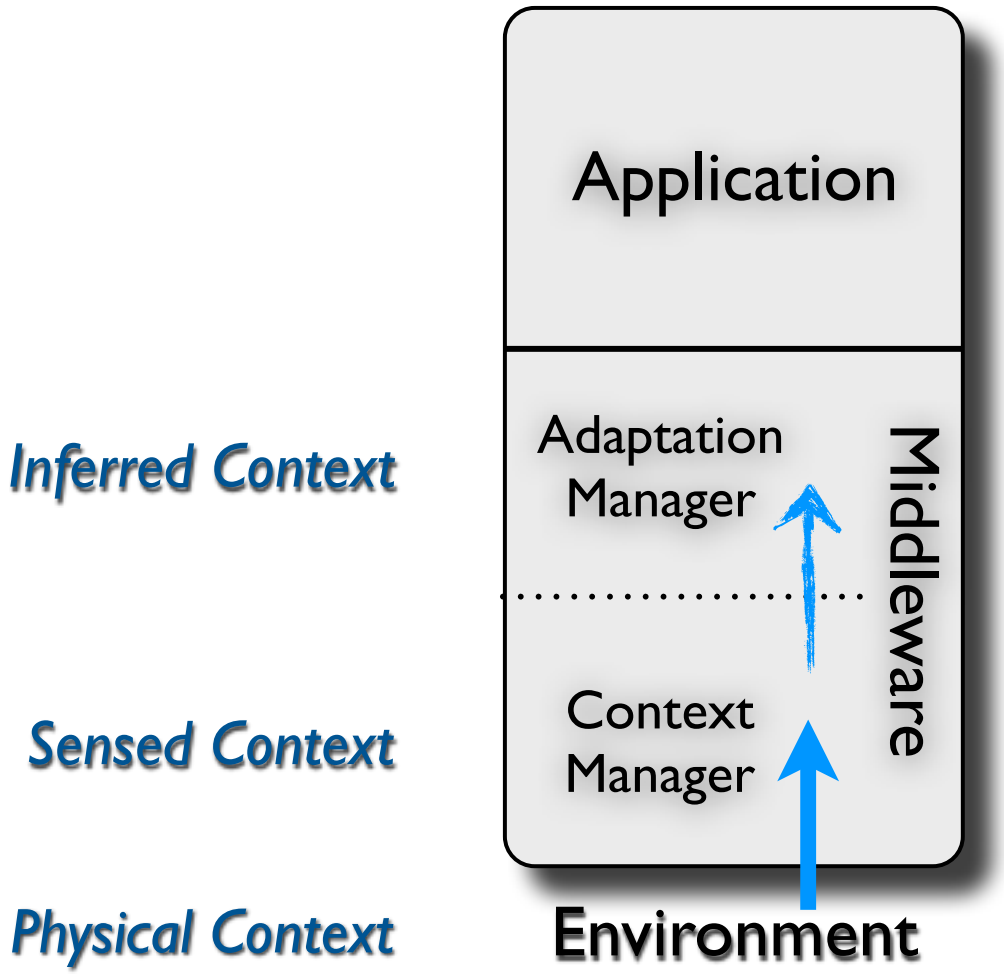
Adaptation in CAAAs



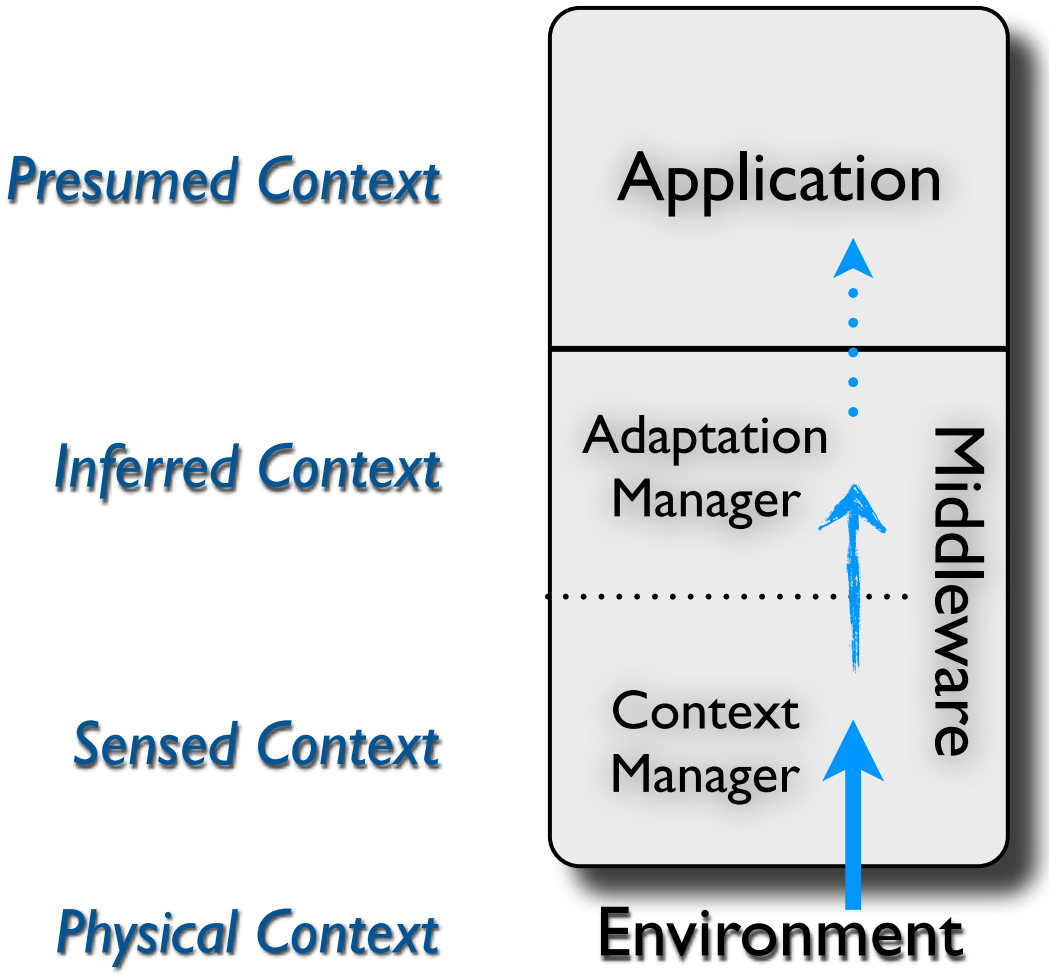
Adaptation in CAAAs



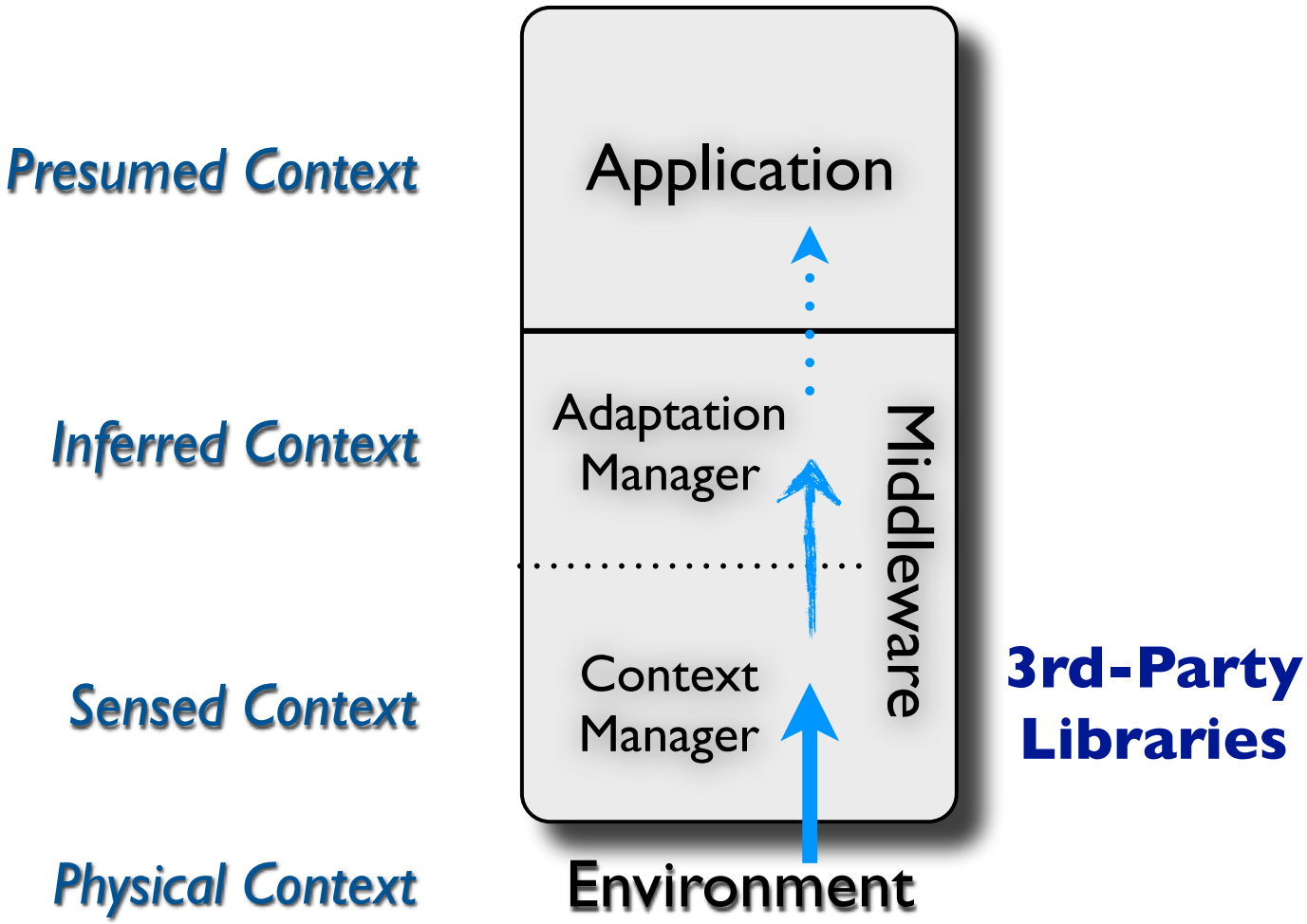
Adaptation in CAAAs



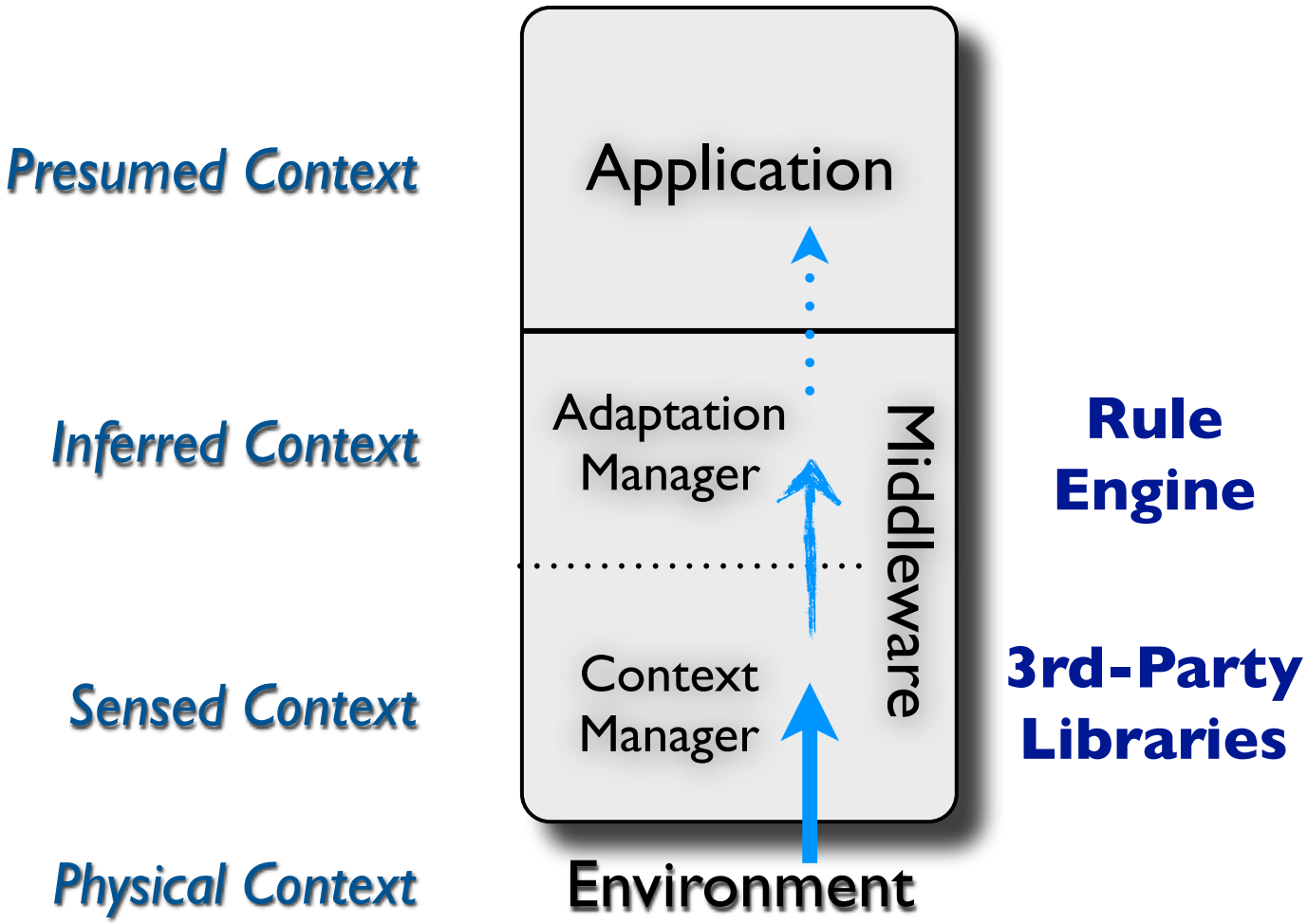
Adaptation in CAAAs



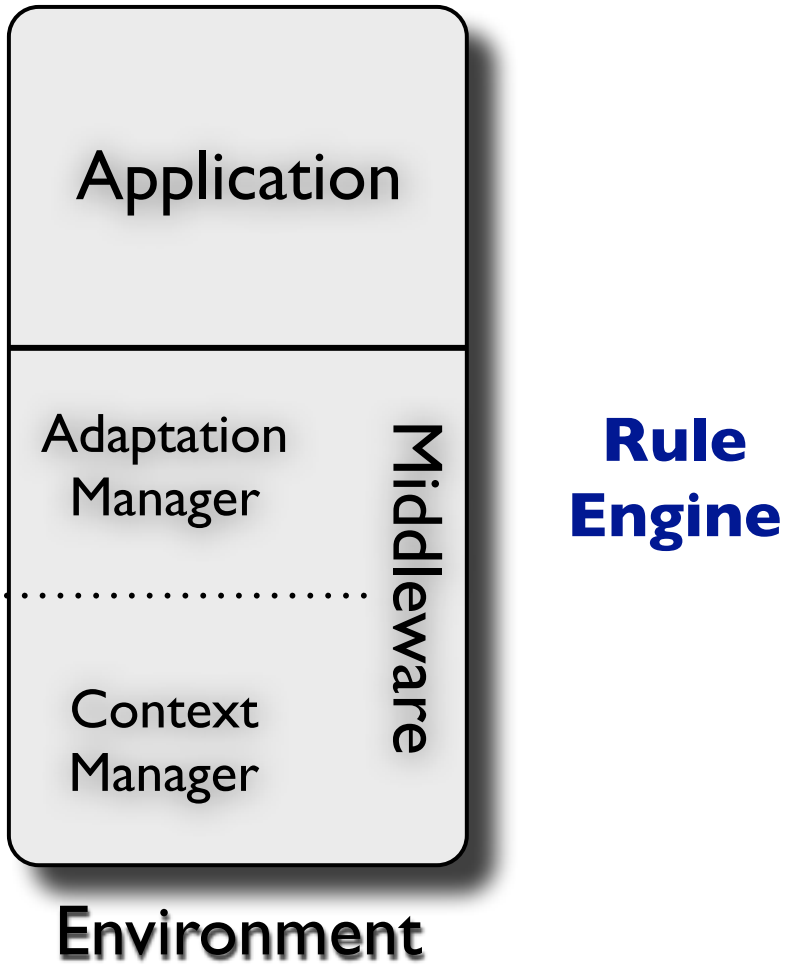
Adaptation in CAAAs



Adaptation in CAAAs



Validation of CAAAs

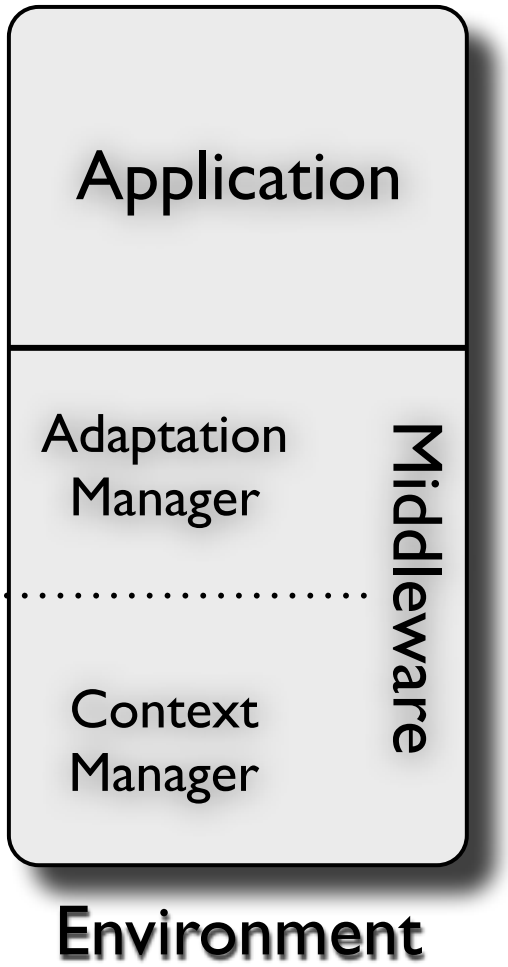


Validation of CAAAs

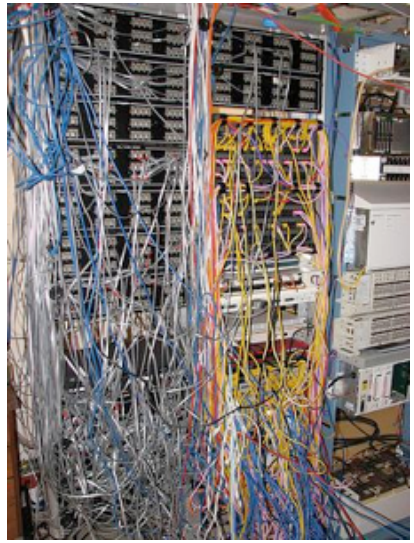
Rules are strongly interdependent

and have multiple priorities

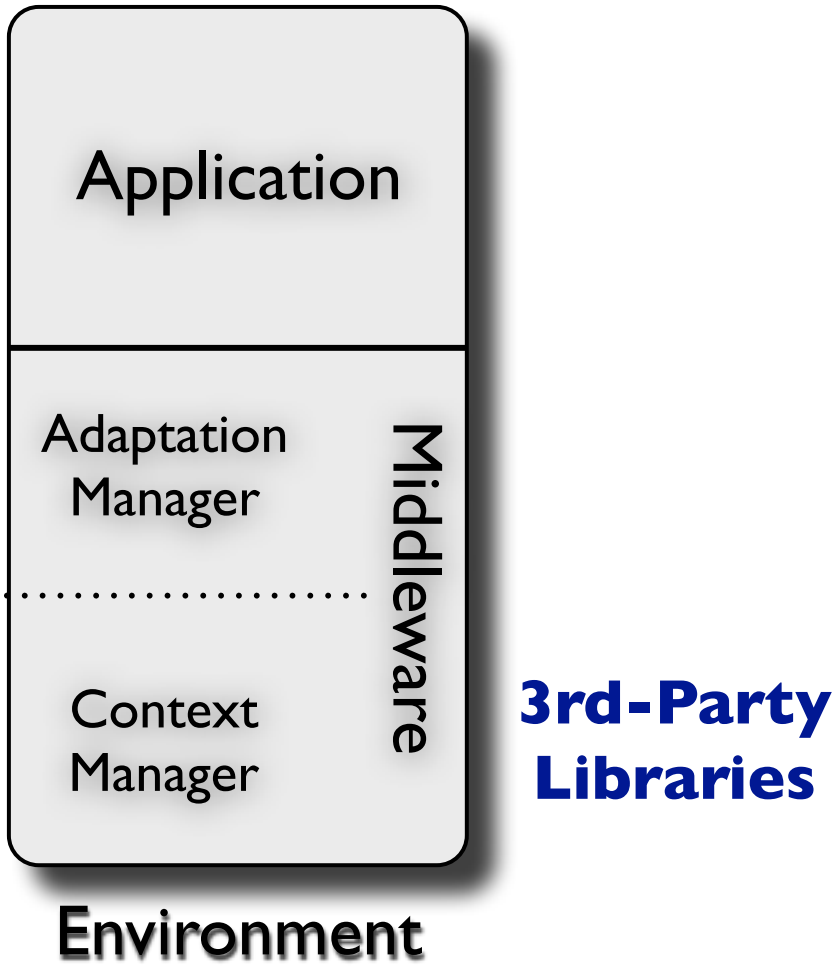
making reasoning difficult even for a small number of rules



Rule Engine



Validation of CAAAs

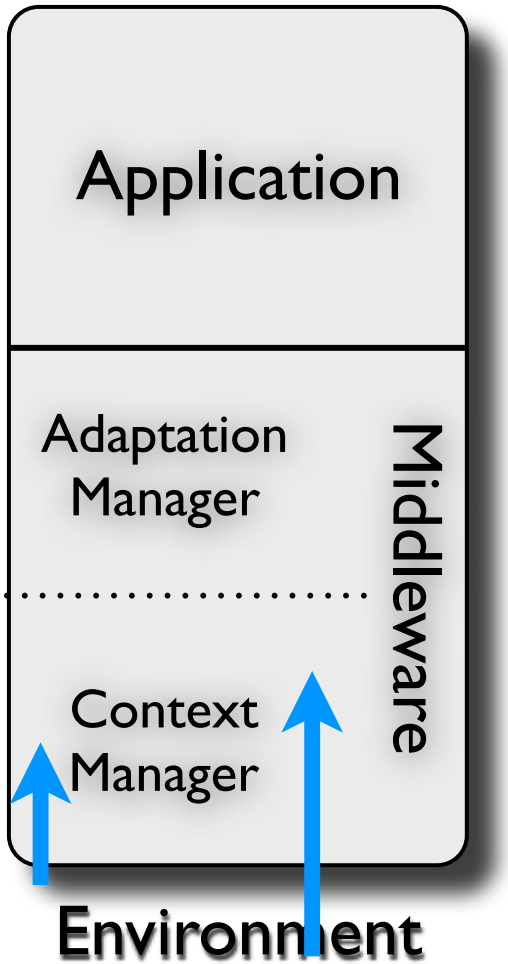


Validation of CAAAs

Context is sensed periodically

from multiple sources

at varying rates



3rd-Party Libraries

Approach

1. Derive *Adaptation Finite-State Machine* (A-FSM) from rule logic
2. Explore state space of A-FSM to discover potential faults
   - ✓ *Enumerative algorithms*
   - ✓ *Symbolic algorithms*
   - ✓ *Planner-based counterexample generation*
3. (Confirm existence of discovered faults)

PhoneAdapter

PhoneAdapter

silent, vibrate

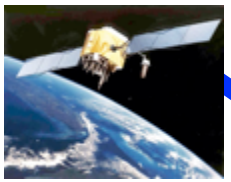
normal, vibrate

loud, divert to hands-free

silent, divert to voicemail

loud, vibrate

PhoneAdapter



silent, vibrate

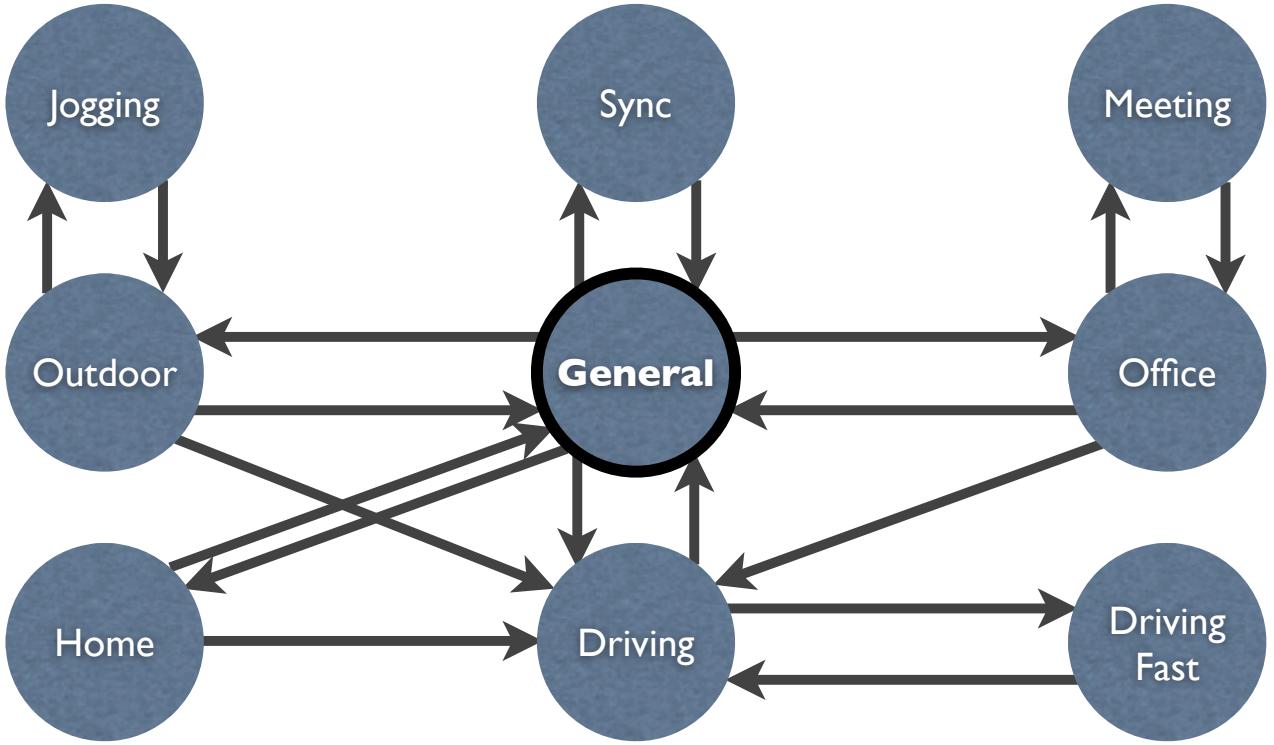
loud, divert to hands-free

normal, vibrate

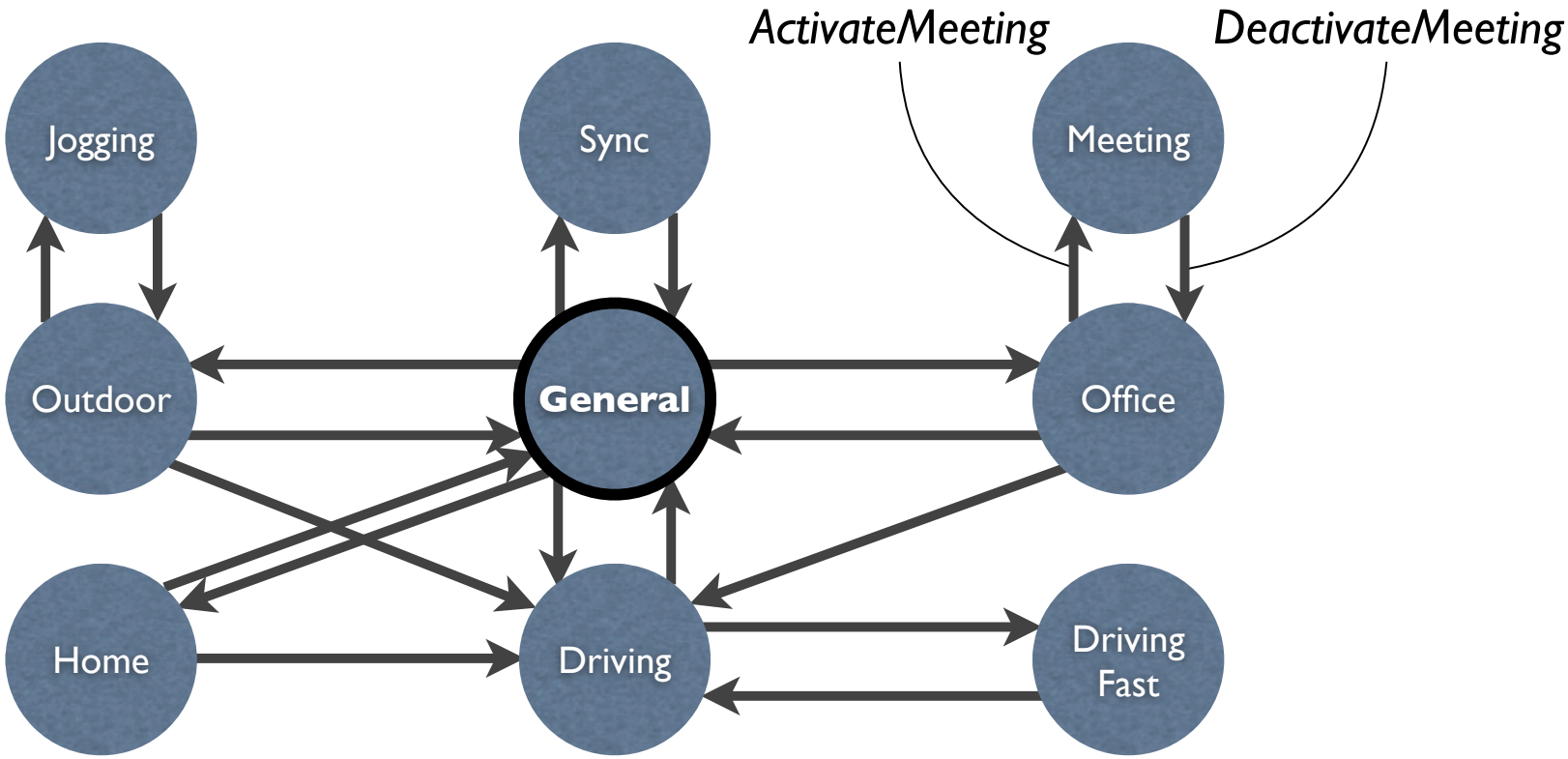
silent, divert to voicemail

loud, vibrate

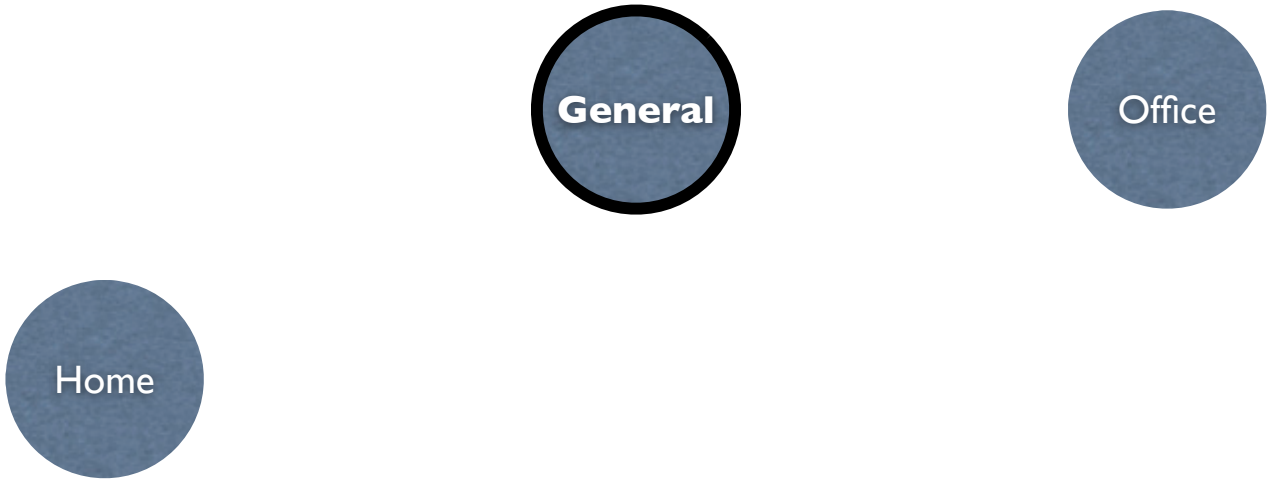
PhoneAdapter A-FSM



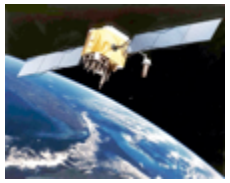
PhoneAdapter A-FSM



Example Faults in PhoneAdapter



Example Faults in PhoneAdapter



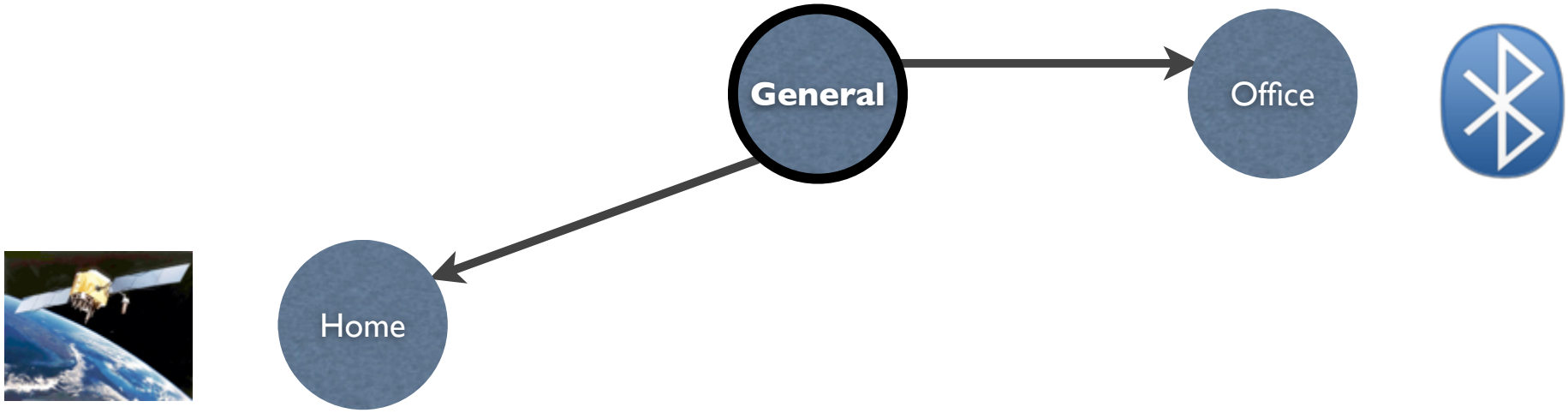
Home

General

Office

User's phone discovers office PC at home (or vice versa)

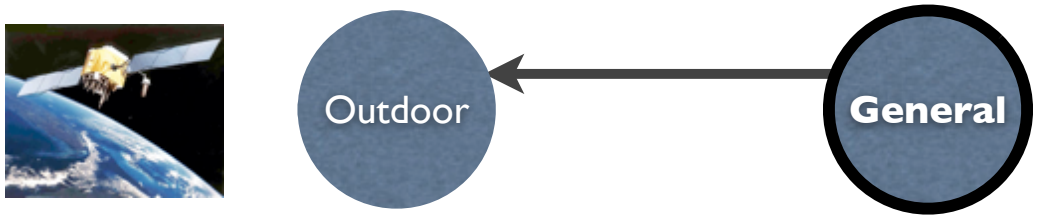
Example Faults in PhoneAdapter



Nondeterminism!

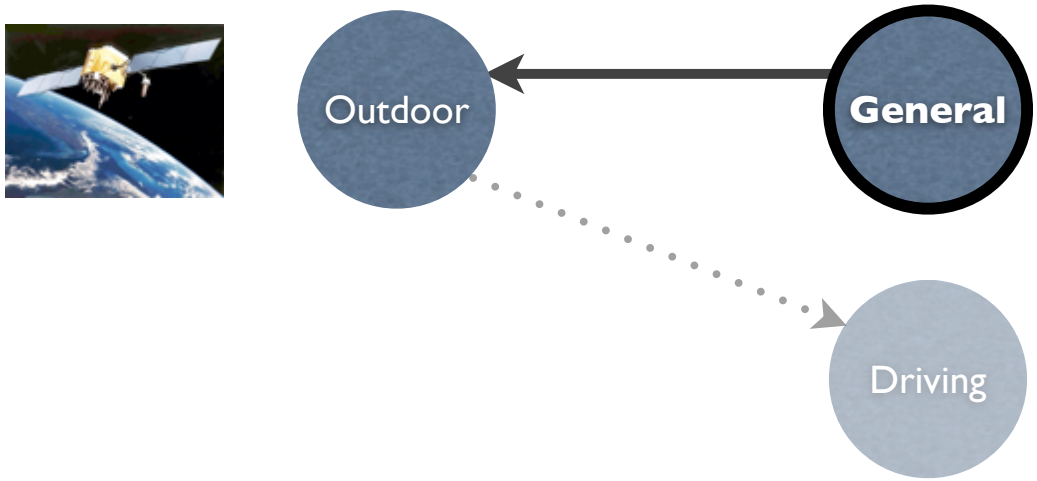
Example Faults in PhoneAdapter

Example Faults in PhoneAdapter



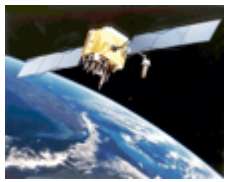
User leaves home

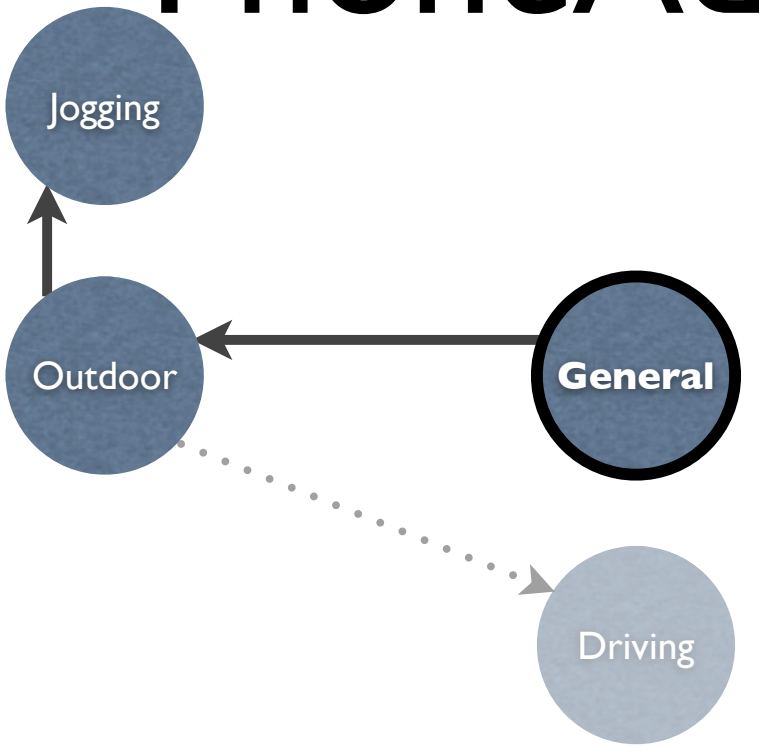
Example Faults in PhoneAdapter



User starts driving before Bluetooth detects hands-free system

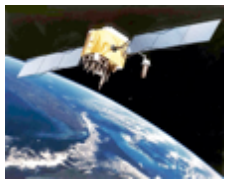
Example Faults in PhoneAdapter

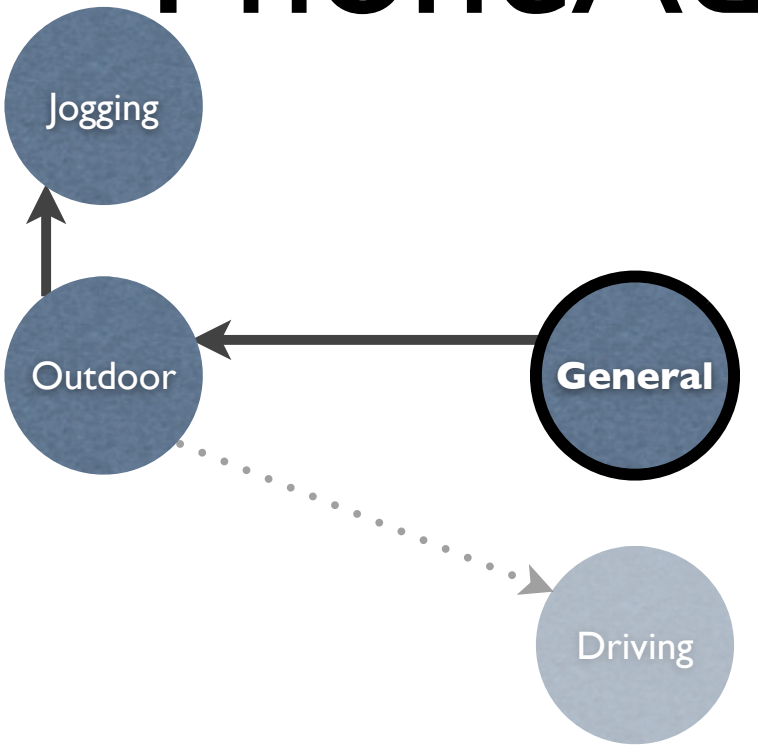




Activation hazard!

Example Faults in PhoneAdapter





Activation hazard!

Faults in CAAAs

- Behavioral Faults
  - *Nondeterminism*
  - *Dead rule*
  - *Dead state*
  - *Unreachable state*
  - *Activation race*
  - *Activation cycle*

Faults in CAAAs

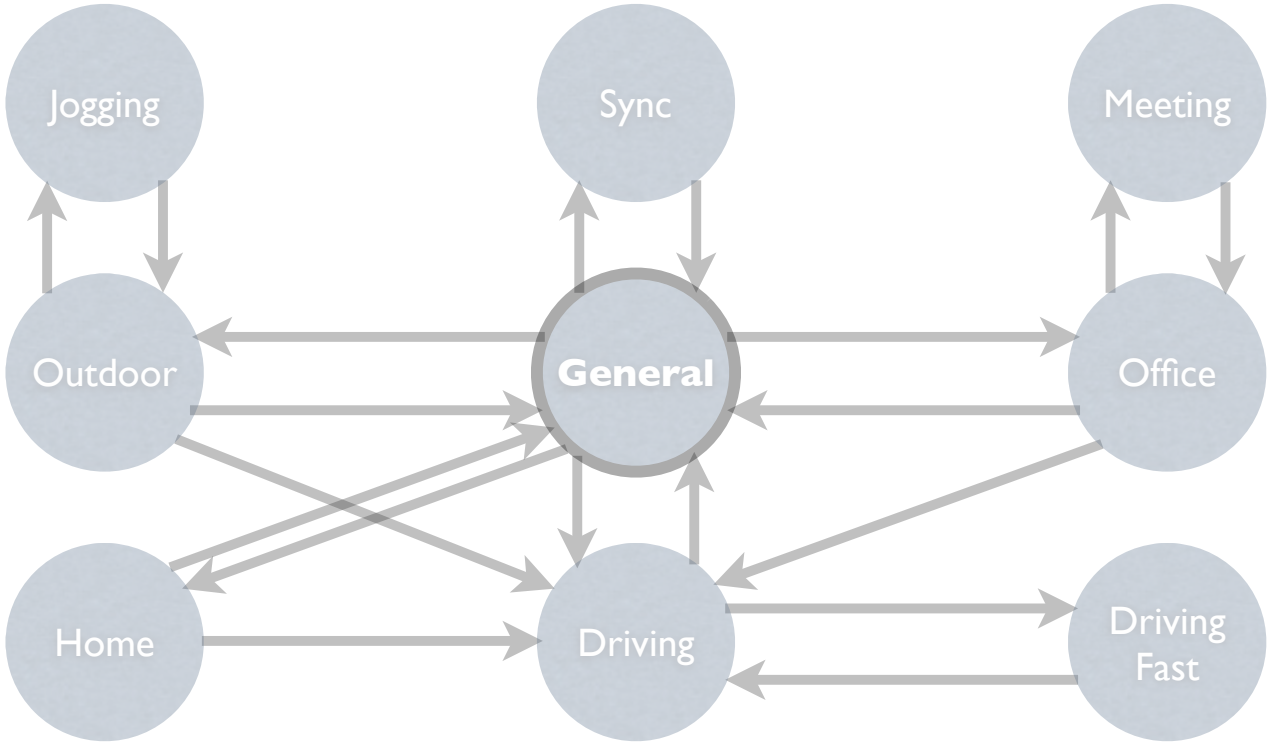
- Behavioral Faults
  - *Nondeterminism*
  - *Dead rule*
  - *Dead state*
  - *Unreachable state*
  - *Activation race*
  - *Activation cycle*
- Hazards
  - *Hold hazard*
  - *Activation hazard*
  - *Priority inversion hazard*

Why Not Use Model Checkers?

- Difficult to encode fault patterns as temporal logic formulae
  - ✳ *Bisimilar models may fail differently*
- Difficult to encode rule logic as models in common model checkers
  - ✳ *Predicates and actions label the transitions*
- Difficult to interpret counterexamples as faults in adaptation behavior

Algorithms

Basic Operation

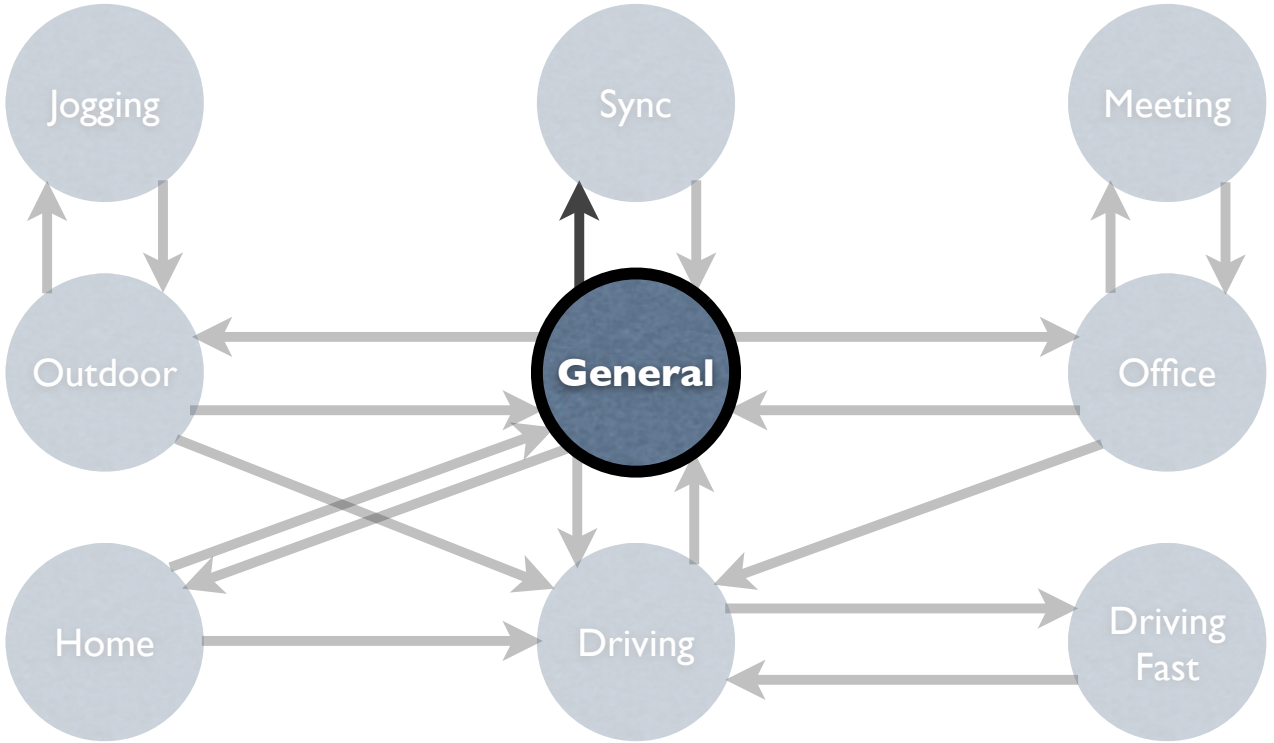


For each state

Analyze rules & inputs, and search along transitions for instances of fault patterns

Algorithms

Basic Operation

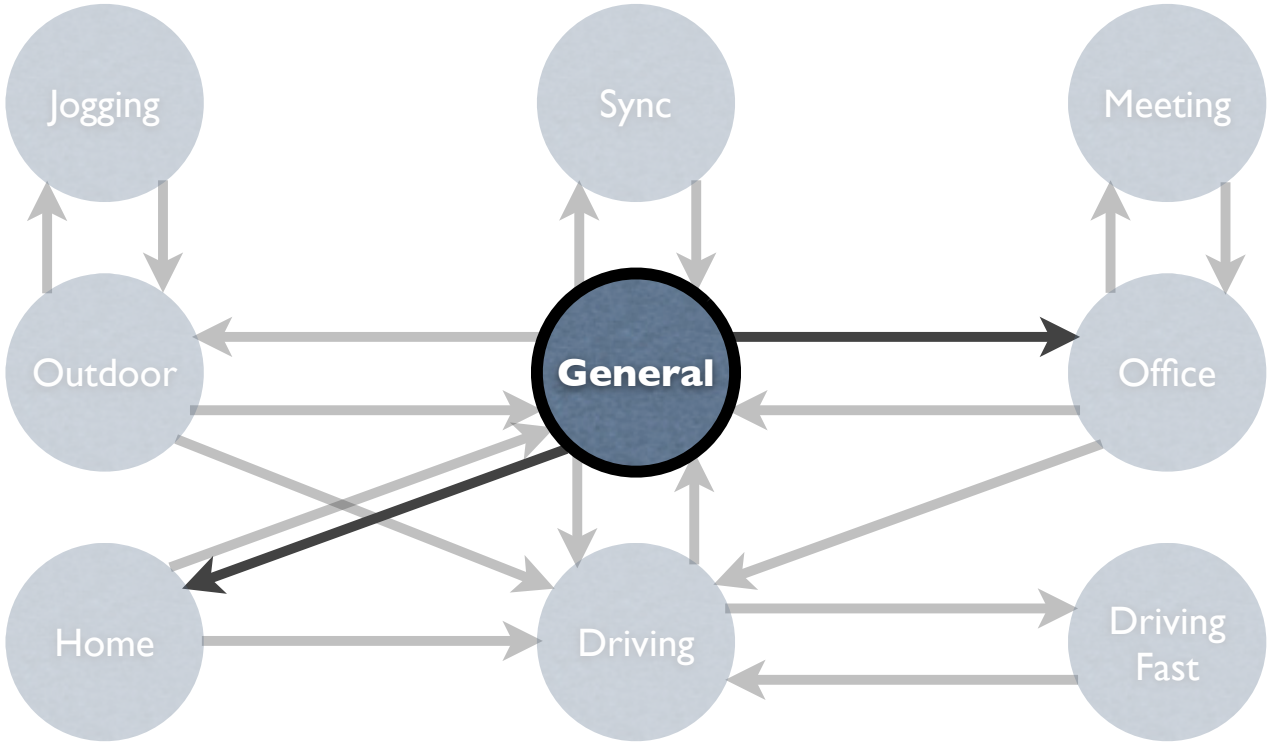


For each state

Analyze rules & inputs, and search along transitions for instances of fault patterns

Algorithms

Basic Operation

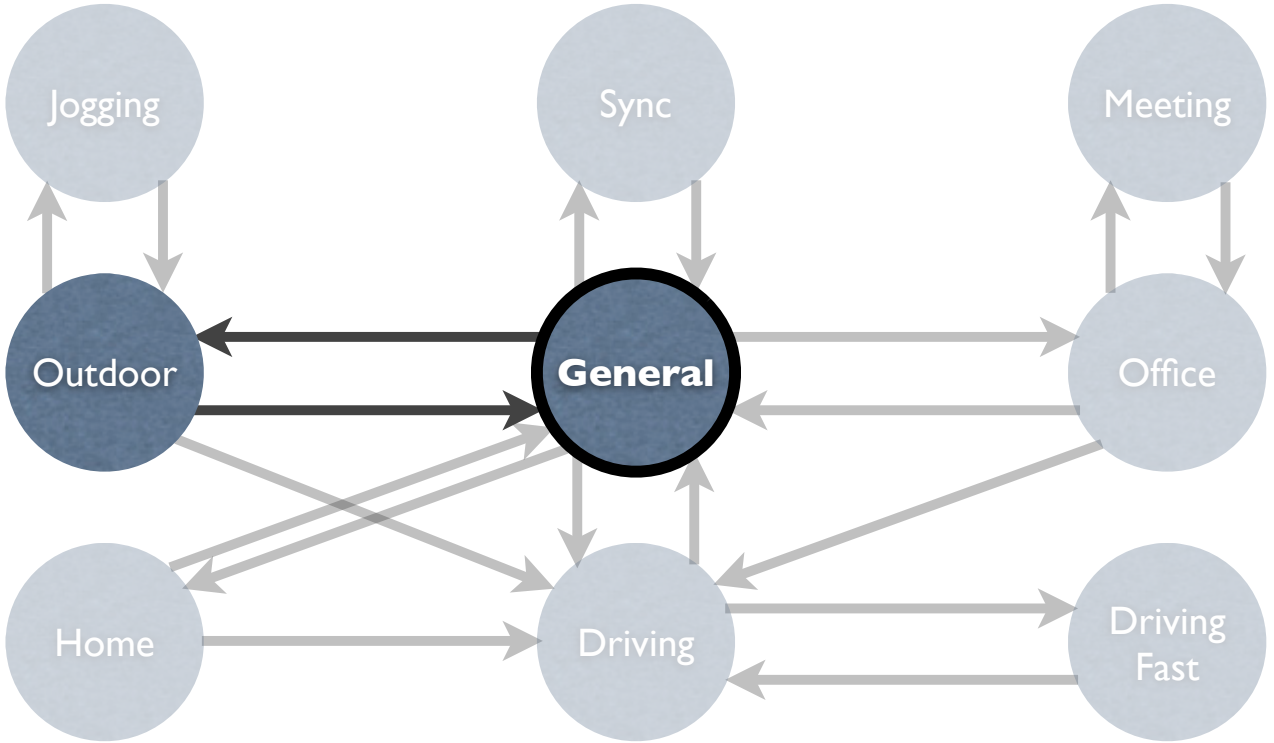


For each state

Analyze rules & inputs, and search along transitions for instances of fault patterns

Algorithms

Basic Operation



For each state

Analyze rules & inputs, and search along transitions for instances of fault patterns

PhoneAdapter Results

Behavioral Faults: Enumerative, Symbolic

State	Nondeterministic Adaptations	Dead Predicates	Adaptation		Unreachable States
			Races	Cycles	
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

PhoneAdapter Results

Hazards: Enumerative

State	Context Hazards			
	Paths	Hold	Activ.	Prior.
General	14085	0	11	3182
Outdoor	161	0	0	52
Jogging	2	0	0	0
Driving	16	2	2	4
DrivingFast	2	0	0	0
Home	104	8	0	13
Office	82634	1828	368	2164
Meeting	0	0	0	0
Sync	2	2	0	0

Conclusion

Comparison of Approaches

Enumerative	Symbolic	Hybrid	Planner
Local Search	*Local Search*	*Local Search*	*Global Search*
Less Precise	*Less Precise*	*Less Precise*	*More Precise*
Concrete Counterexamples	*Symbolic Counterexamples*	*Symbolic Counterexamples*	*Concrete Counterexamples*
Handles Smaller State Spaces	*Handles Big State Spaces*	*Handles Bigger State Spaces*	*Sequential Search*
Fast	*Faster*	*Fastest*	*Slowest*

Future Work

Verification

- Continue the work on hazards and planners
- Quantitative reasoning about faults
  - Battery level, movement timings, etc.
- Online analysis of rules and faults

Future Work

Design

- *Alternatives to rule-based adaptation!*
  - Machine learning approaches to context classification and adaptation selection

Future Work

Design

- *Alternatives to rule-based adaptation!*
  - Machine learning approaches to context classification and adaptation selection

- **Felicitous Computing Institute**

Thank You!

REFERENCES

Z. Wang, S. Elbaum and D.S. Rosenblum, *Automated Generation of Context-Aware Tests*, **Proc. 2007 Int'l Conf. on Software Engineering** (ICSE 2007), Minneapolis, MN, USA, May 2007, pp. 406–415.

M. Sama, D.S. Rosenblum, Z. Wang and S. Elbaum, *Multi-Layer Faults in the Architectures of Mobile, Context-Aware Adaptive Applications: A Position Paper*, Short Paper, **Proc. ICSE 2008 Workshop on Software Architectures and Mobility** (SAM 2008), Leipzig, Germany, May 2008, pp. 47–49.

M. Sama, F. Raimondi, D. Rosenblum and W. Emmerich, *Algorithms for Efficient Symbolic Detection of Faults in Context-Aware Applications*, **Proc. 1st Int'l Workshop on Automated Engineering of Autonomous and Run-Time Evolving Systems** (ARAMIS 2008), L'Aquila, Italy, Sep. 2008, pp. 1–8.

M. Sama, D.S. Rosenblum, Z. Wang and S. Elbaum, *Model-Based Fault Detection in Context-Aware Adaptive Applications*, **Proc. 16th ACM SIGSOFT Int'l Symposium on the Foundations of Software Engineering** (FSE 2008), Atlanta, GA, USA, Nov. 2008, pp. 261–271.

J. Cubo, F. Raimondi, M. Sama and D. Rosenblum, *A Model to Design and Verify Context-Aware Adaptive Service Composition*, **Proc. IEEE Int'l Conf. on Services Computing** (SCC 2009), Bangalore, India, Sep. 2009, pp. 184–191.

M. Sama, D.S. Rosenblum, Z. Wang and S. Elbaum, *Multi-Layer Faults in the Architectures of Mobile, Context-Aware Adaptive Applications*, **Journal of Systems and Software**, invited paper for Special Issue on Software Architecture and Mobility, Vol. 83, Issue 6, Jun. 2010, pp. 906–914.

M. Sama, S. Elbaum, F. Raimondi and D.S. Rosenblum, *Context-Aware Adaptive Applications: Fault Patterns and Their Automated Identification*, **IEEE Transactions on Software Engineering**, invited paper for Special Issue on the Best Papers of FSE 2008, Vol. 36, No. 5, Sep./Oct. 2010, pp. 644–661.