MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

Model- Driven Development of High-Assurance Dynamically Adaptive Systems

Betty H.C. Cheng,

Software Engineering and Network Systems Lab
Digital Evolution Laboratory
BEACON: NSF Center for Evolution in Action
Department of Computer Science and Engineering
Michigan State University
chengb at cse dot msu dot edu
http://www.cse.msu.edu/~chengb

MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

Acknowledgments

Current and Former Students:

- Andres Ramirez
- Adam Jensen
- Chad Byers
- Ji Zhang
- Masoud Sadjadi
- Heather Goldsby
- Ben Beckmann
- David Knoester

Sponsors

- U.S. Army Research Office
- U.S. Department of the Navy
- National Science Foundation
- Michigan State University

Additional Collaborators

- Philip McKinley (CSE)
- Charles Ofria (CSE)
- Xiaobo Tan (ECE)
- Raza Haque (Medicine)

Industrial Collaborators:

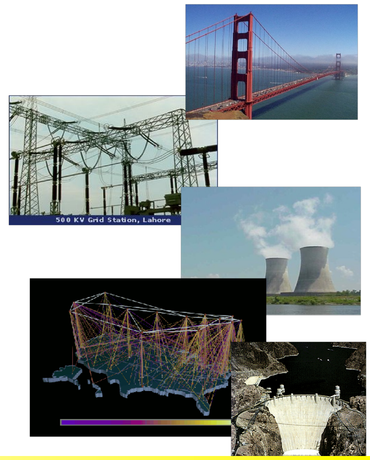
- Ford Motor Research
- General Motors
- HP
- Siemens
- Motorola
- Continental Automotive

MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

High-Assurance Autonomic Computing

- **Autonomic computing:** Promises self-managed and long-running systems with limited human guidance.
- Systems must continue to ***operate correctly*** during exceptional situations, upgrades, and evolution under ***uncertain*** conditions
- Need for assurance
  - hardware component failures
  - network outages
  - software faults
  - security attacks



ONR CIP RAPIDware project revealed many open challenges for current and future systems

MICHIGAN STATE UNIVERSITY

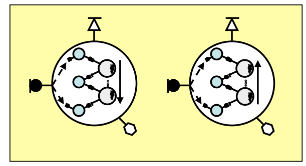
Software Engineering & Network Systems Lab

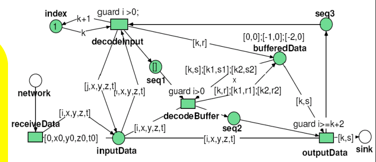
RAPIDware Project

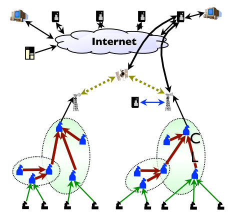
- Funded by U.S Office of Naval Research
  - Adaptable Software / Critical Infrastructure Protection Program
  - 2001-2007
- High-assurance software adaptation technologies for:
  - Detecting and responding to environmental changes
  - Strengthening self-auditing capabilities of "always-on" systems
- Autonomic cyberinfrastructure for:
  - crisis management and public safety systems
  - military command and control
  - environmental monitoring
  - management of large industrial installations
- Enable systems to operate *through* failures and attacks

RAPIDware Focus Areas

- **Mechanisms for software adaptation**
  - Language extensions, middleware tools [Wear02,IEEE04, ICAC04, DOA04, DARE04, DEAS05,SPE06,]
  - Decision-making [ICDL04, TKDE07]
  - Migration path for existing software [WOSS02, Auton06, MiSE07]
- **Assurance in adaptation**
  - Requirements engineering [REFSQ05,SEAMS06,REV07]
  - Formal Analysis [WADS05, JSS06, ISSE07]
  - Design technologies [ICSE06]
  - Run-time support [WADS04,M@RT07,MiSE07]
- **Adaptive cyberinfrastructure**
  - Adaptive group security
  - Extension to sensor networks
  - "Sensor to server" coordination

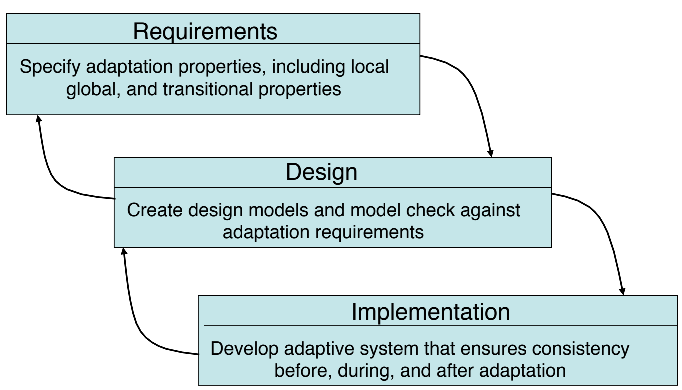






RAPIDware Approach

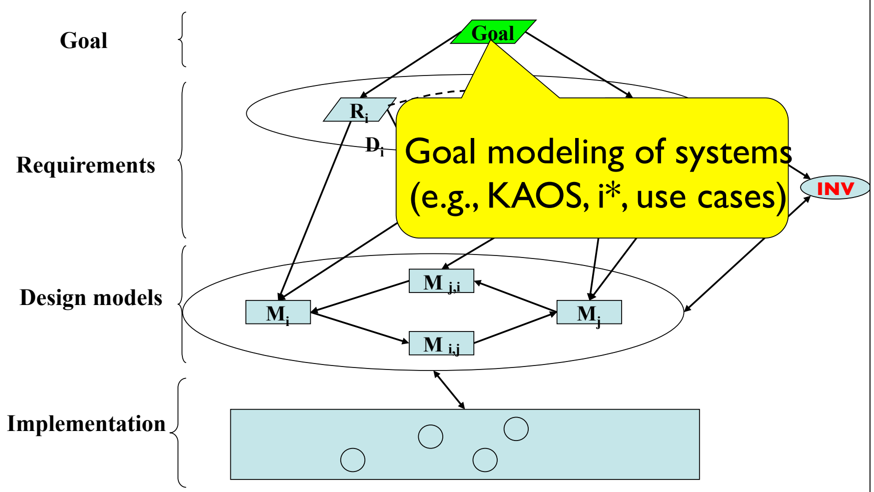
- Gain assurance in adaptive software through
  - A systematic software development process
  - Application of formal methods at every stage



MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

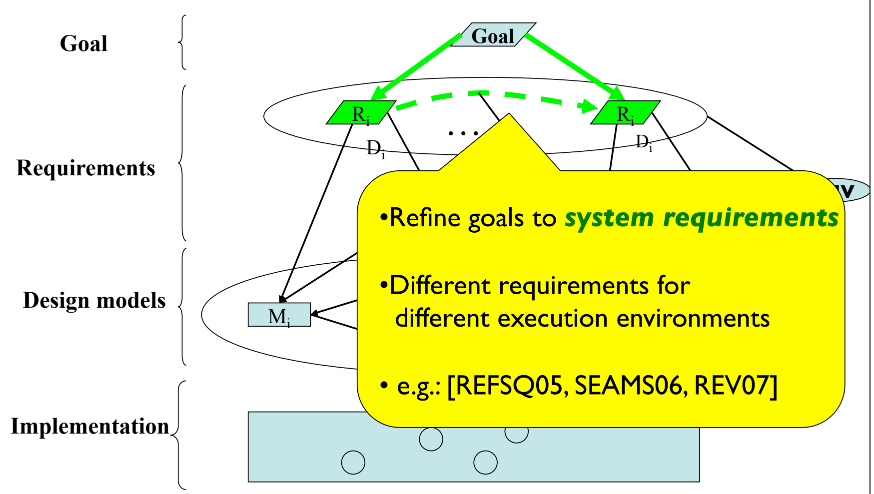
System Goal

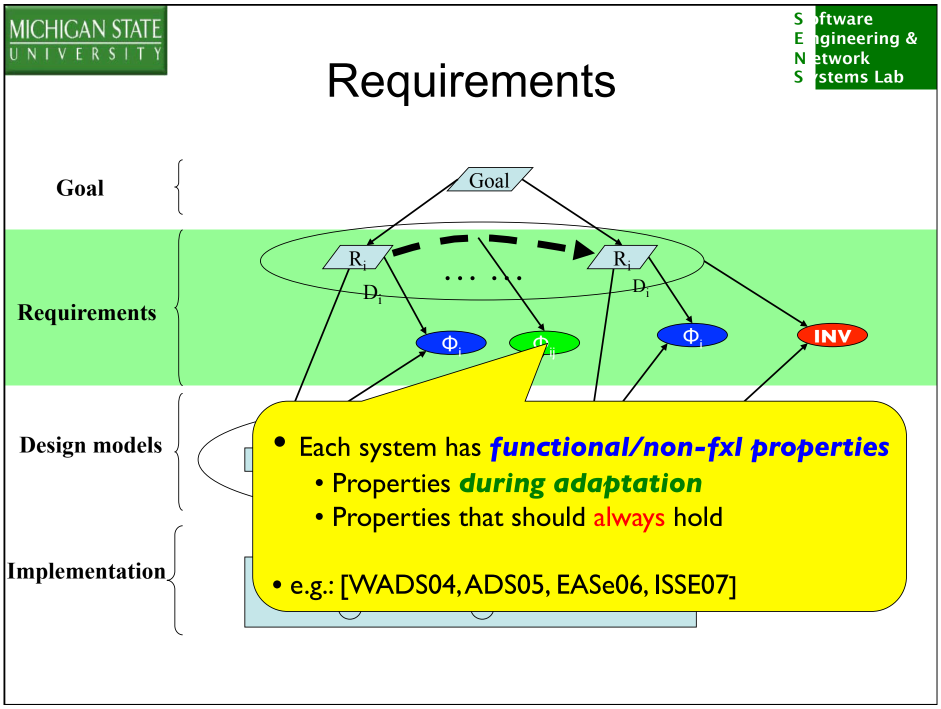


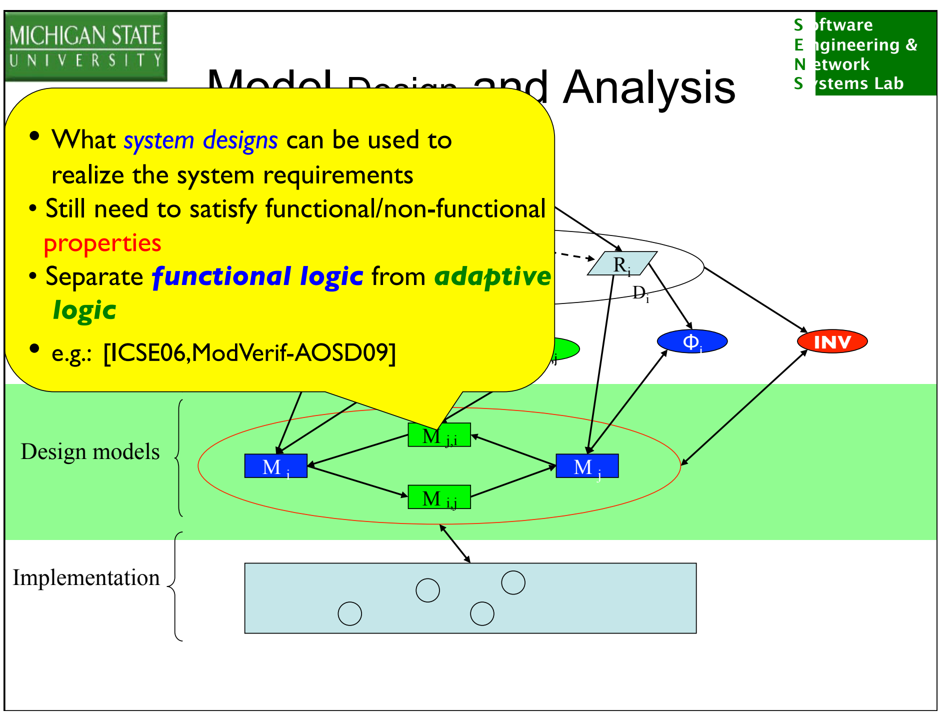
MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

Goal Refinement



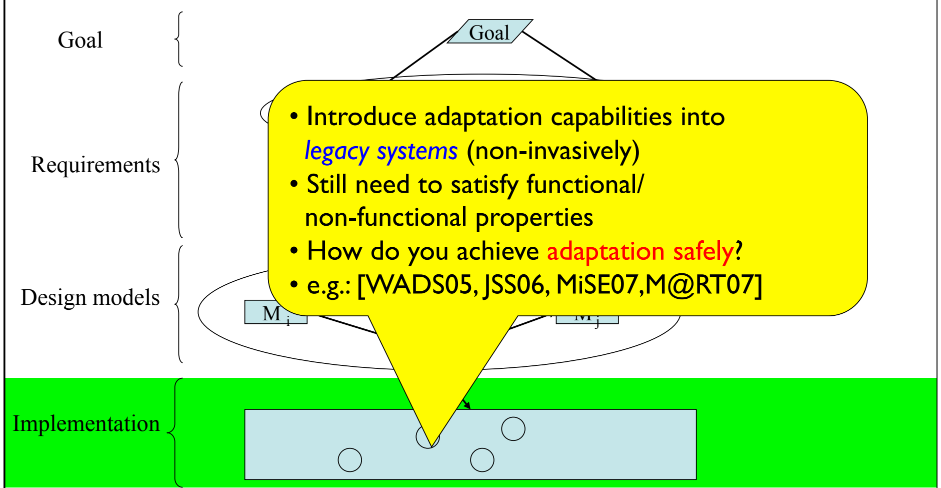

MICHIGAN STATE
UNIVERSITY
Software
Engineering &
Network
Systems Lab
Requirements
Goal
Requirements
Design models
Implementation
Goal
INV
• Each system has functional/non-fxl properties
• Properties during adaptation
• Properties that should always hold
• e.g.: [WADS04, ADS05, EASe06, ISSE07]


MICHIGAN STATE
UNIVERSITY
Software
Engineering &
Network
Systems Lab
Model Design and Analysis
• What system designs can be used to realize the system requirements
• Still need to satisfy functional/non-functional properties
• Separate functional logic from adaptive logic
• e.g.: [ICSE06,ModVerif-AOSD09]
INV
Design models
Implementation

MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

Implementation



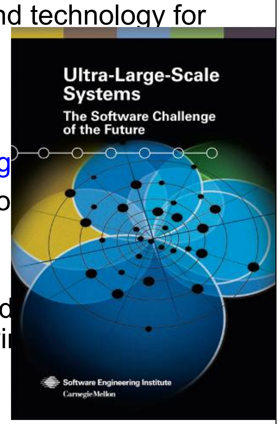
- Introduce adaptation capabilities into *legacy systems* (non-invasively)
- Still need to satisfy functional/ non-functional properties
- How do you achieve adaptation safely?
- e.g.: [WADS05, JSS06, MiSE07,M@RT07]

MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

Emerging Area

- SEI conducted study on the need for research and technology for Ultra-Large Scale software systems
  - (report by SEI, 2006: L. Northrup *et al)*
  - http://www.sei.cmu.edu/uls/
- Objective: Support Future Information Processing
  - Superior collection, fusion, analysis, and use o meet future computational objectives
- Requires: increasingly complex systems
  - Thousands of platforms, sensors, decision nod systems connected through heterogeneous wi networks.
- New scale:
  - Data stored, accessed, manipulated, and refined
  - Number of connections/interdependencies among components
  - Number of hardware elements



MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

New Scale

Ultra-Large Scale SW-Intensive Systems

Healthcare Infrastructure

Software Engineering Institute | Carnegie Mellon

Ultra-Large-Scale Systems
Linda Northrop, ICSE 2007
© 2007 Carnegie Mellon University

20

MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

New Scale

High-Assurance Cyberphysical Systems

Intelligent Transportation and Vehicle Systems

Software Engineering Institute | Carnegie Mellon

Ultra-Large-Scale Systems
Linda Northrop, ICSE 2007
© 2007 Carnegie Mellon University

21

Context: "*Sufficient*" System Health

High-level Objective:

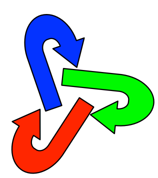
– ***How to design a safe adaptive system with incomplete information and evolving environmental conditions***

Challenges:

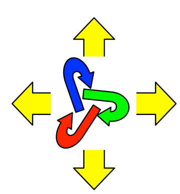
- **Execution environment**
  - **How to model environment**
  - **How to effectively monitor changing conditions**
  - **Adaptive monitoring**
- **Decision-making for dynamic adaptation**
  - Decentralized control
  - Assurance guarantees (functional and non-functional constraints)
- **Adaptation mechanisms:**
  - Application level
  - Middleware level

The ULS Ecosystem

- Key elements:
  - Computing devices
  - Business and organizational policies
  - Environment (including people)



- Forces:
  - Competition for resources
  - Unexpected environmental changes
  - Decentralized control
  - Demand for assurance



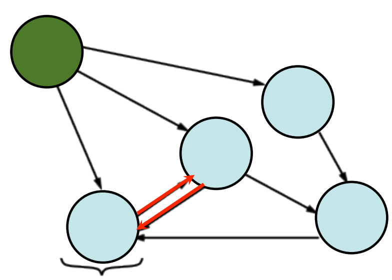
uncertainty is an inherent problem with ULS systems...

Dynamic Adaptation and **Autonomic Computing** are enabling technologies to help systems evolve to respond to changing conditions...

But what (combinations of) conditions should be considered?

Motivation

- To develop an autonomic system:
  - Identify **target systems**
    - Non-adaptive systems
    - i.e., Business logic
  - Adaptations among steady-state systems
    - Adapt from source to target system



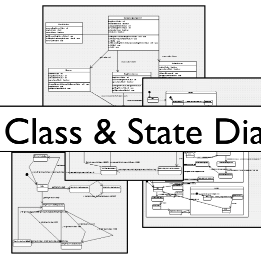
Viable Target System

MICHIGAN STATE UNIVERSITY

S oftware E ngineering & N etwork S ystems Lab

Motivation

- To develop an autonomic system:
  - Identify target systems
    - Non-adaptive systems
    - i.e., Business logic
  - Adaptations among steady-state systems
    - Adapt from source to target system
  - Model multiple viable target systems using UML *(defacto standard)*



UML Class & State Diagrams

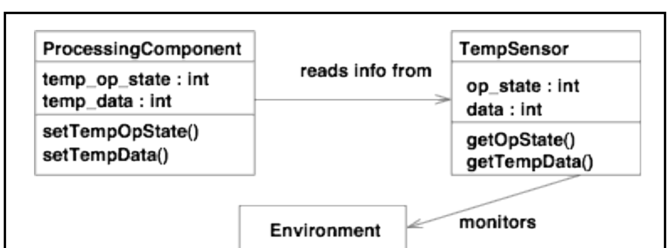
MICHIGAN STATE UNIVERSITY

MICHIGAN STATE UNIVERSITY

S oftware E ngineering & N etwork S ystems Lab

Use UML to Model Structure

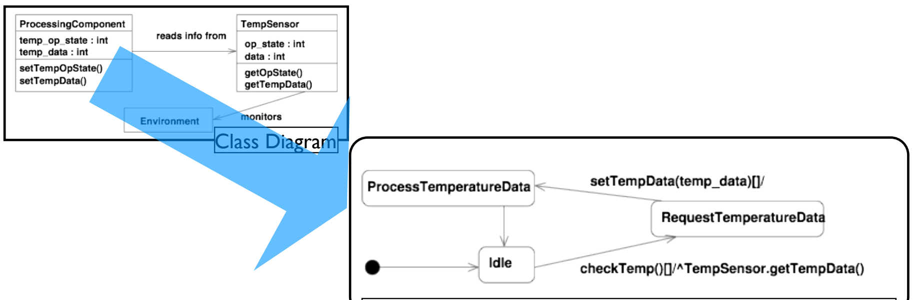
- UML Class Diagram
  - Identifies the system *elements (classes)*
    - E.g., `ProcessingComponent`, `TempSensor`, `Environment`
  - Describes the *relationships* between elements (classes)
    - E.g., `ProcessingComponent reads info from TempSensor` and `TempSensor monitors Environment`



Class Diagram

Use UML to Model Behavior

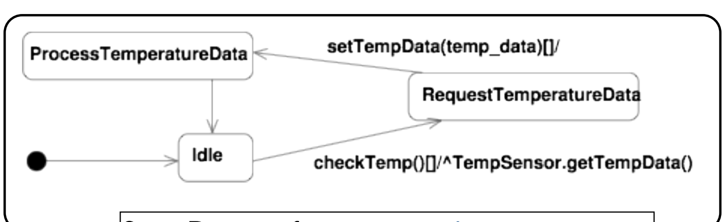
- UML State Diagram
  - Describes the *behavior of an element (class)*
  - One state diagram per element (class) on the class diagram
    - E.g., 3 state diagrams -- one for `ProcessingComponent`, one for `TempSensor`, and one for `Environment`



State Diagram for **`ProcessingComponent`**

Objective: Automatic Generation of Behavior Models

- UML State Diagram
  - **States:**
    - E.g., `Idle, RequestTemperatureData, ProcessTemperatureData`
  - **Transitions** - triggers[guards]/actions
    - E.g., `checkTemp()[]/^TempSensor.getTempData()`, `setTempData(temp_data)[]/`



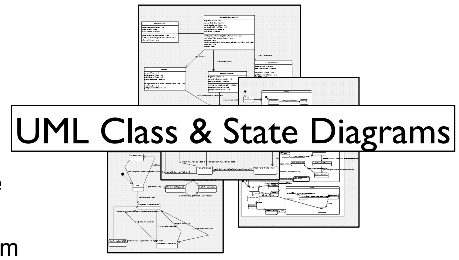
State Diagram for **`ProcessingComponent`**

MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

Motivation

- To develop an autonomic system:
  - Identify target systems
    - Non-adaptive systems
    - i.e., Business logic
  - Adaptations among steady-state systems
    - Adapt from source to target system
  - Model multiple viable target systems using UML *(defacto standard)*

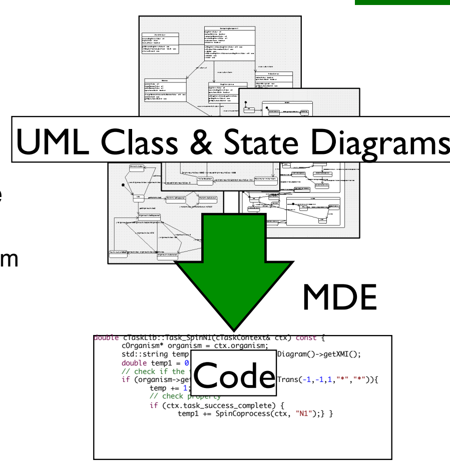


MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

Motivation

- To develop an autonomic system:
  - Identify target systems
    - Non-adaptive systems
    - i.e., Business logic
  - Adaptations among steady-state systems
    - Adapt from source to target system
  - Model multiple viable target systems using UML *(defacto standard)*
  - Leverage Model-Driven Engineering (MDE)

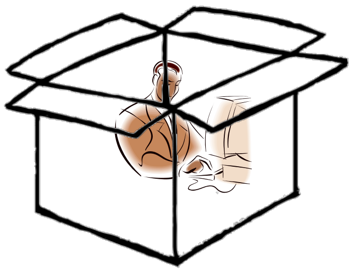


Challenge: construct viable target system models that address uncertainty in environment...

MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

Now for something outside the box …



MICHIGAN STATE UNIVERSITY

Inspiration from Nature

Software Engineering & Network Systems Lab

- Living organisms have amazing ability to adapt to changing environments
  - Short term: phenotypic plasticity
    - (e.g., snails change tooth shape depending environment)
  - Long
    - (e.g
- Most c traits de self-ma
- Examp
  - Syster
  - Recon)

Can evolution be leveraged to identify viable target systems?

Bio-Inspired Work

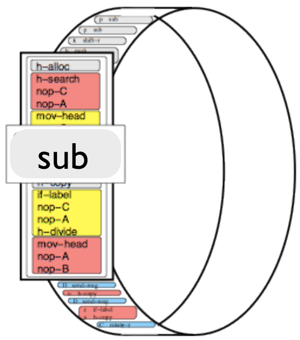
- Biomimetic systems
  - *Mimic* behavior of natural systems (e.g., swarm UAVs)
- Evolutionary computation
  - Must: *mutate, replicate*, and *compete*
- 3 types of evolutionary computation
  - **Genetic Algorithms:** Individuals encode a solution
  - **Genetic Programming**: Individuals are programs
  - **Digital evolution** [Science2004, Nature1999, Nature2003]
    - Individuals are programs
    - Individuals are self-replicating
    - Receives more resources to replicate faster by performing ***tasks***
    - No explicit fitness function -- evolution is more open-ended
    - Individuals can interact with each other

Approach: Digital Evolution

- Avida
  - Digital evolution platform
  - Used to conduct pioneering research in evolution of biocomplexity [Lenski03Nature, Wilke01Nature, Wilke04Science]

Approach: Digital Evolution

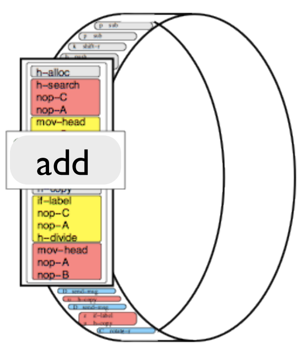
– Organism = computer program



An Organism

Approach: Digital Evolution

– Organism = computer program
- Genome (instruction) mutations



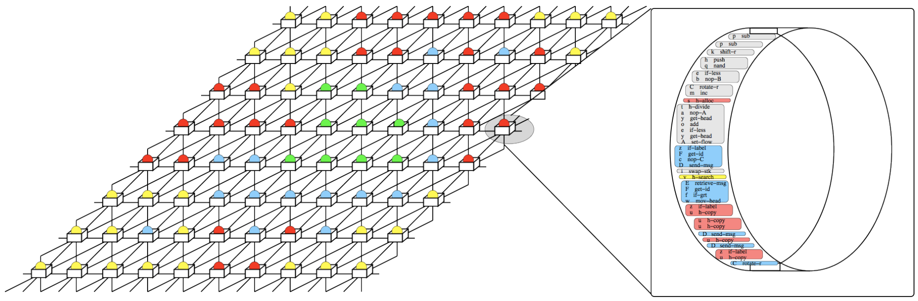
An Organism

MICHIGAN STATE UNIVERSITY

Approach: Digital Evolution

Software Engineering & Network Systems Lab

– Organism = computer program
  - Genome (instruction) mutations
  - Exist in a computational environment

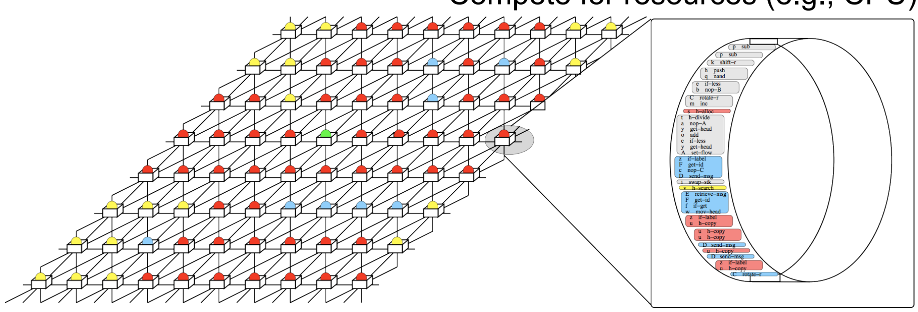


MICHIGAN STATE UNIVERSITY

Approach: Digital Evolution

Software Engineering & Network Systems Lab

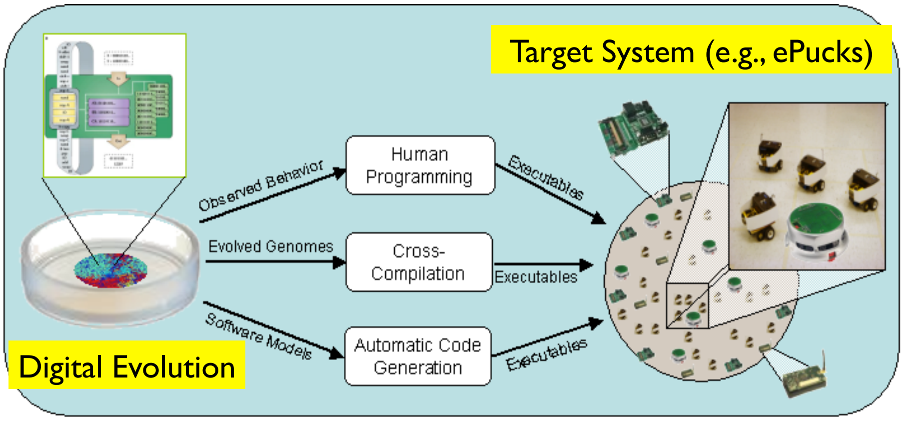
– Organism = computer program
  - Genome (instruction) mutations
  - Exist in a computational environment
  - Compete for resources (e.g., CPU)



MICHIGAN STATE UNIVERSITY

Designing Software with Digital Evolution

Software Engineering & Network Systems Lab



- *IEEE Computer*, "Harnessing Digital Evolution", Jan 2008.

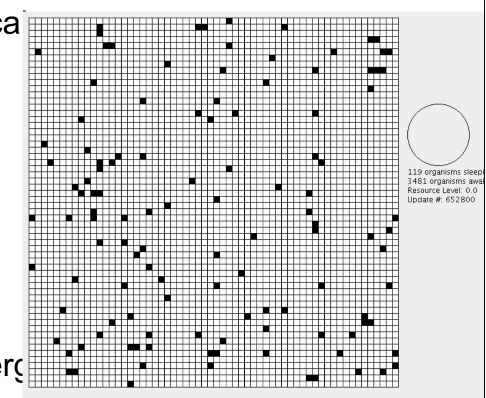
MICHIGAN STATE UNIVERSITY

Energy Management

[ECAL07b,SASO-07]

Software Engineering & Network Systems Lab

- Energy management is critical issue in mobile devices
- Use Avida to investigate the evolution of energy conservation
- Organisms can conserve energy by executing sleep instructions
- Organisms that manage energy efficiently will proliferate

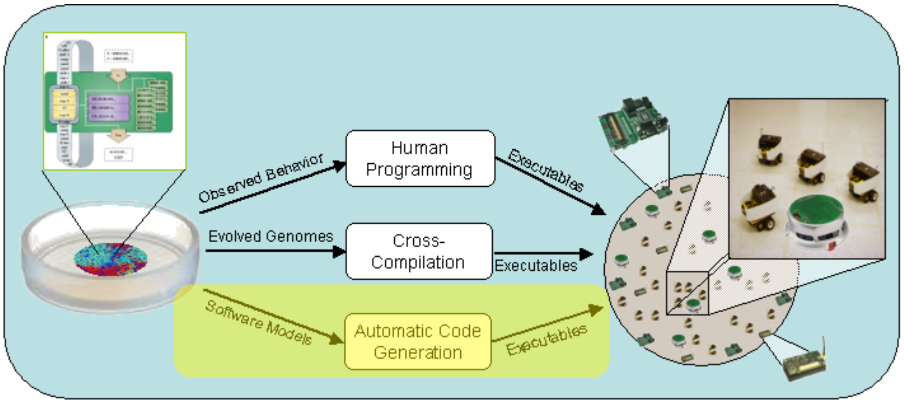


"Early to bed, early to rise…"

MICHIGAN STATE UNIVERSITY

Designing Software with Digital Evolution

Software Engineering & Network Systems Lab



MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

ORCHID: Harnessing Digital Evolution to Develop High-Assurance Dynamically Adaptive Systems

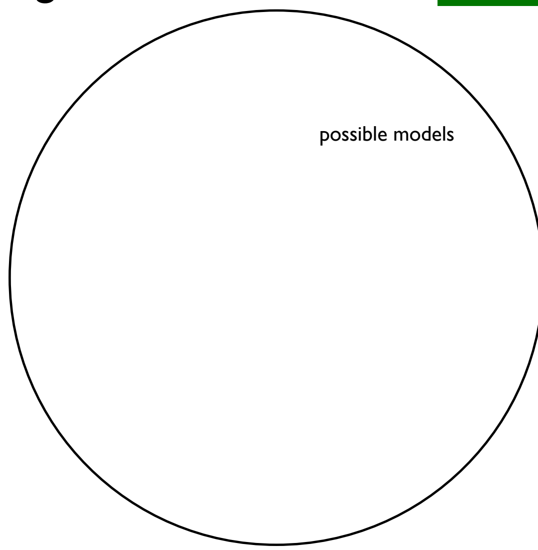
40

MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

Big Picture

- Objective: *Generate behavioral models*
  - *Resilient behavior*
  - *Potentially innovative*

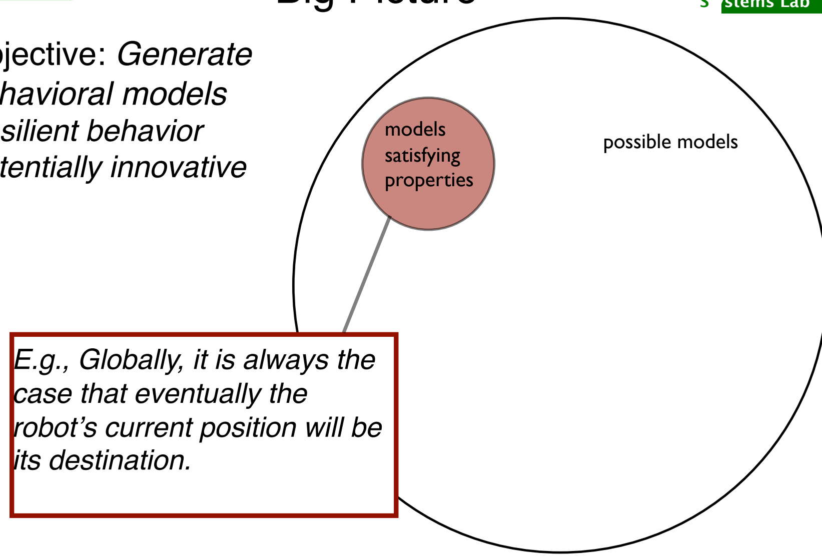


MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

Big Picture

- Objective: *Generate behavioral models*
  - *Resilient behavior*
  - *Potentially innovative*

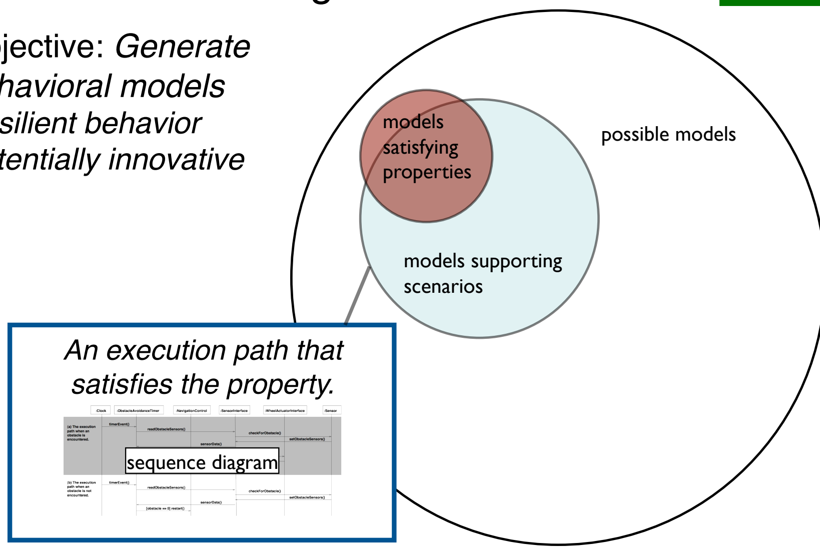


MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

Big Picture

- Objective: *Generate behavioral models*
  - *Resilient behavior*
  - *Potentially innovative*

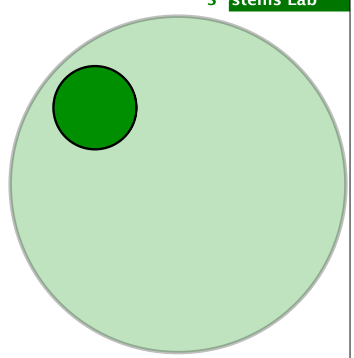


MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

Technique: Brute Force Generation

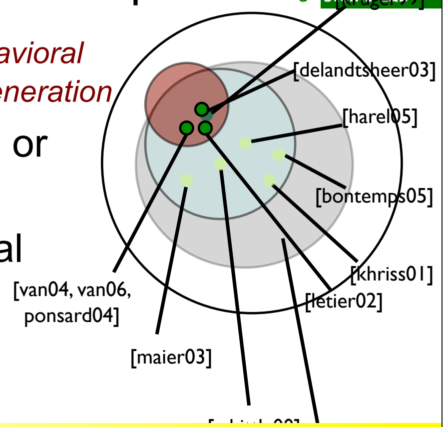
- Purpose: *generate all possible models to find solution*
- Input: Alphabet, properties
- Output: Multiple behavioral models
- Characteristics:
  - Alphabet: defined by triggers, guards and actions
  - Generates every possible model
  - Discovers ALL models that satisfy properties
  - *Not actually feasible with limited resources (e.g., time, CPU)*



- **Light green = generated model**
- **Dark green = solution**

Synthesis Techniques

- Purpose: *Generate a behavioral model for analysis & code generation*
- Input: Basic scenarios or properties
- Output: One behavioral model

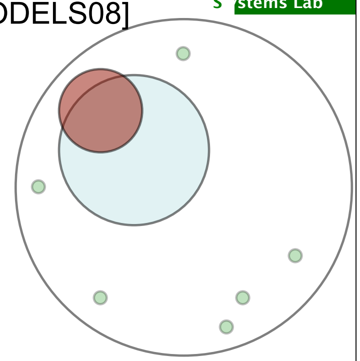


Synthesis techniques based on specific design strategy or algorithm -- limited by what is known at development time

Avida-MDE

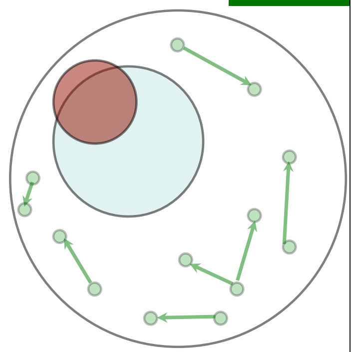
[ICAC08,GECCO-08,MODELS08]

- Start with a population of organisms that generate models



Avida-MDE [ICAC08,GECCO-08]

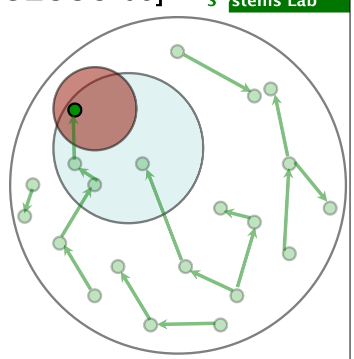
- Start with a population of organisms that generate models
- Organisms replicate and are mutated
  - Different models are produced



- Arrows indicate parent-child relationship
- Direction indicates "better/worse" model

Avida-MDE [ICAC08,GECCO-08]

- Start with a population of organisms that generate models
- Organisms replicate and are mutated
  - Different models are produced
- Selection pressures (tasks) drive evolution:



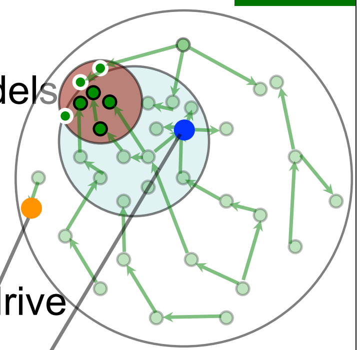
- Model density increases closer to the solution space

MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

Avida-MDE [ICAC08,GECCO-08]

- Start with a population of organisms that generate models
- Organisms replicate and are mutated
  - Different models are produced
- Selection pressures (tasks) drive evolution:
  - *Poor performing organisms die*
  - ***Better organisms have more children***
  - E.g., scenarios, properties, software engineering metrics
    - Deterministic states, fewer transitions

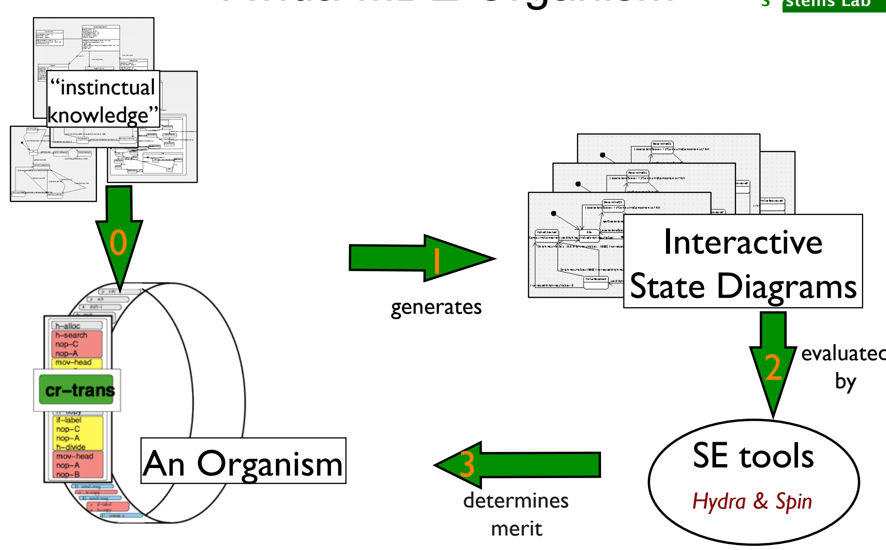


- **White outlined solutions satisfy properties, but do not support all scenarios**

MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

Avida-MDE Organism

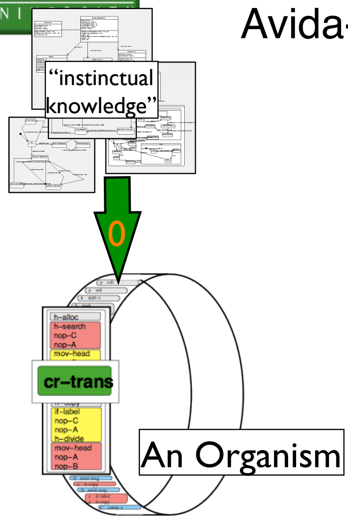


Skip

MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

Avida-MDE Organism



- **Instinctual knowledge** - Information embedded in an organism at birth
  - Class diagram information (*alphabet*)
    - triggers (methods)
    - guards (boolean expressions of attributes)
    - actions (methods of associated classes)
  - Existing state diagrams (if any)

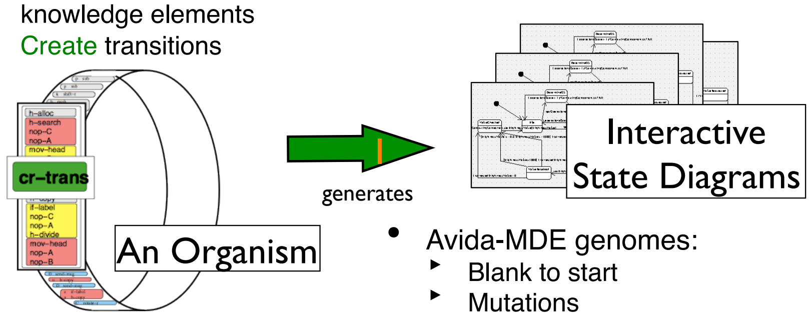
How does an organism use its instinctual knowledge to generate state diagrams?

MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

Avida-MDE Organism

- Possible instructions:
  - Select instinctual knowledge elements
  - Create transitions



- Avida-MDE genomes:
  - Blank to start
  - Mutations
    - Insert/remove/swap instructions
    - Offspring state diagram ≠

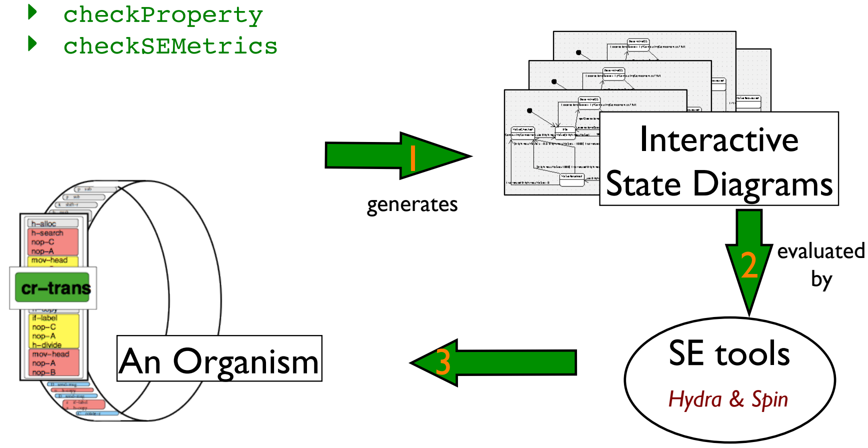
How does Avida-MDE assess the generated state diagrams?

MICHIGAN STATE UNIVERSITY

Avida-MDE Organism

Software Engineering & Network Systems Lab

- Avida-MDE Tasks (Selection Pressures):
  - checkScenarios
  - checkProperty
  - checkSEMetrics

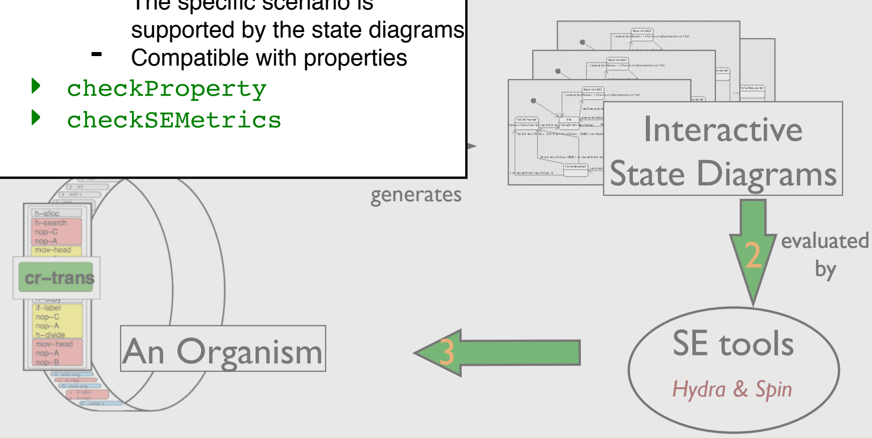


MICHIGAN STATE UNIVERSITY

Avida-MDE Organism

Software Engineering & Network Systems Lab

- Avida-MDE Tasks:
  - checkScenarios
    - The specific scenario is supported by the state diagrams
    - Compatible with properties
  - checkProperty
  - checkSEMetrics

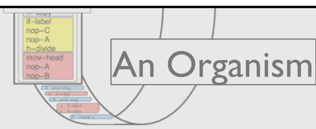


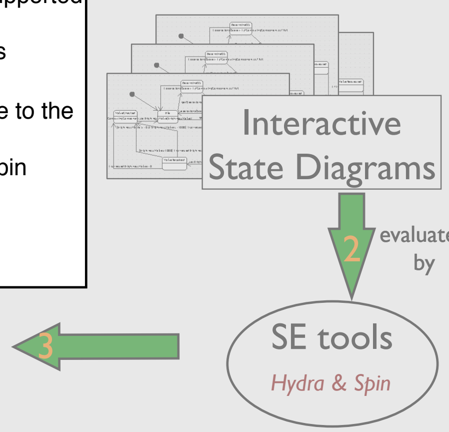
MICHIGAN STATE UNIVERSITY

Avida-MDE Organism

Software Engineering & Network Systems Lab

- Avida-MDE Tasks:
  - `checkScenarios`
    - The specific scenario is supported by the state diagrams
    - Compatible with properties
  - `checkProperty`
    - The state diagrams adhere to the temporal logic property
    - Uses Hydra [ICSE01] & Spin [Holzmann]
  - `checkSEMetrics`



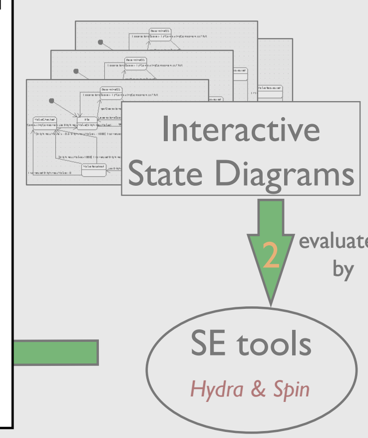


MICHIGAN STATE UNIVERSITY

Avida-MDE Organism

Software Engineering & Network Systems Lab

- Avida-MDE Tasks:
  - `checkScenarios`
    - The specific scenario is supported by the state diagrams
    - Compatible with properties
  - `checkProperty`
    - The state diagrams adhere to the temporal logic property
    - Uses Hydra [ICSE01] & Spin [Holzmann]
  - `checkSEMetrics`
    - Minimum number of transitions used
    - Percentage of states that are deterministic



MICHIGAN STATE UNIVERSITY

Avida-MDE Uses in Software Development

Software Engineering & Network Systems Lab

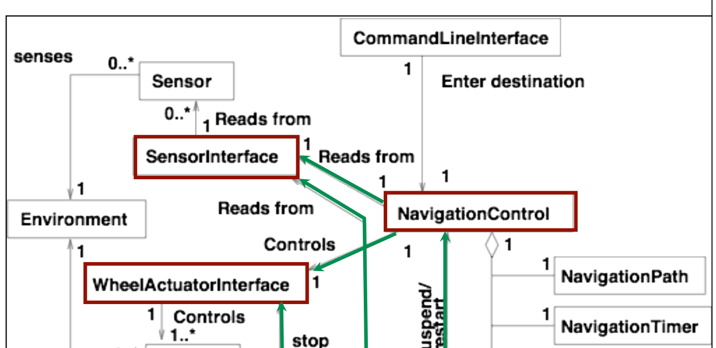
Initial development	New Class extension	New behavior for existing class
• Limited information is available	• A new class is added to an existing system	• Existing classes with new functionality
• Developer provides: ‣ **Class diagram** ‣ **Scenarios**	• Developer provides: ‣ **Class diagram** ‣ ***Some*** **state diagrams** ‣ **Scenarios** ‣ **Properties**	• Developer provides: ‣ **Class diagram** ‣ **State diagrams** ‣ **Scenarios** ‣ **Properties**
• Avida-MDE generates: ‣ A set of state diagrams potentially innovative initial target system	• Avida-MDE generates: ‣ A new state diagram for the new class ‣ Extend existing state diagrams	• Avida-MDE generates: ‣ Extend existing state diagrams to support new behavior

MICHIGAN STATE UNIVERSITY

Case Study: Robot Navigation System

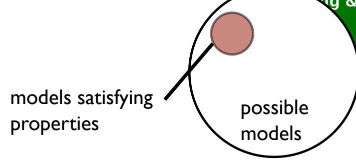
Software Engineering & Network Systems Lab

- T-Rot - Intelligent robot for elderly [Park06ICSE]
  - Autonomously navigate to a destination
  - **Avoid obstacles (*New Class Extension*)**



We describe the desired behavior using scenarios, properties, and software engineering metrics…

Property Tasks



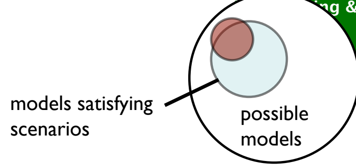
1. Normal Behavior:

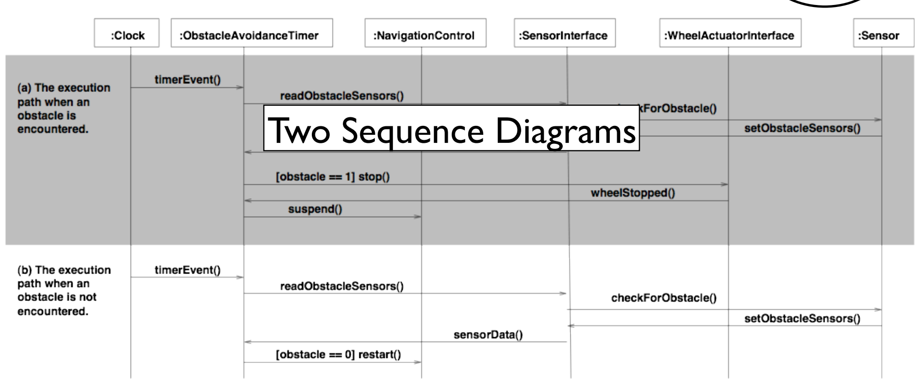
Globally, it is always the case that eventually the robot's current position will be its destination.

- New Behavior:

Globally, it is always the case that if the `ObstacleAvoidanceTimer` *detects an obstacle, then eventually the* `WheelActuatorInterface` *will stop the wheels and the* `NavigationControl` *will be suspended.*

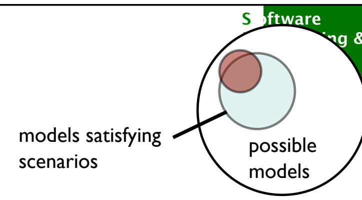
Scenario Tasks





MICHIGAN STATE UNIVERSITY

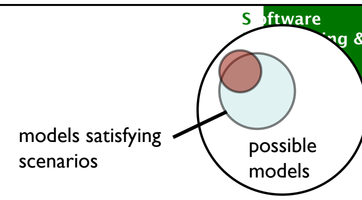
Scenario Tasks



1. When an `obstacle` is detected, the `ObstacleAvoidance` timer stops the `wheels` and `suspends` the `NavigationControl`.

MICHIGAN STATE UNIVERSITY

Scenario Tasks



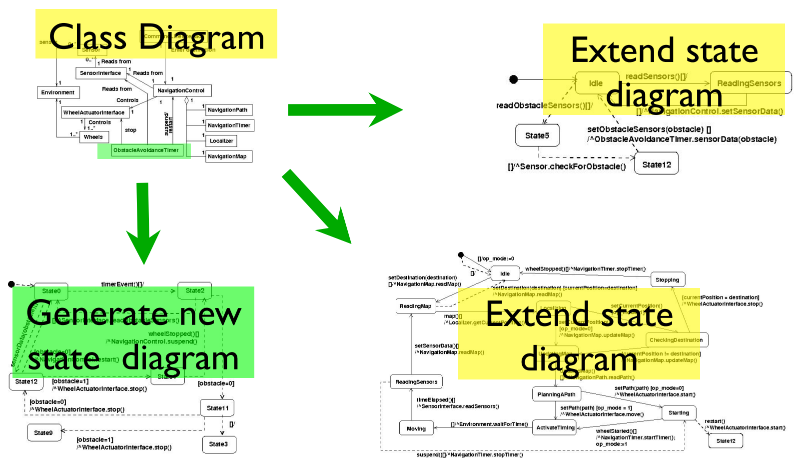
1. When an `obstacle` is detected, the `ObstacleAvoidance` timer stops the `wheels` and `suspends` the `NavigationControl`.
2. When an `obstacle` is not detected, the `ObstacleAvoidance` timer `restarts` the `NavigationControl`.

Results: organisms generated 7 compliant behavioral models.

MICHIGAN STATE UNIVERSITY

Case Study: Robot Control System

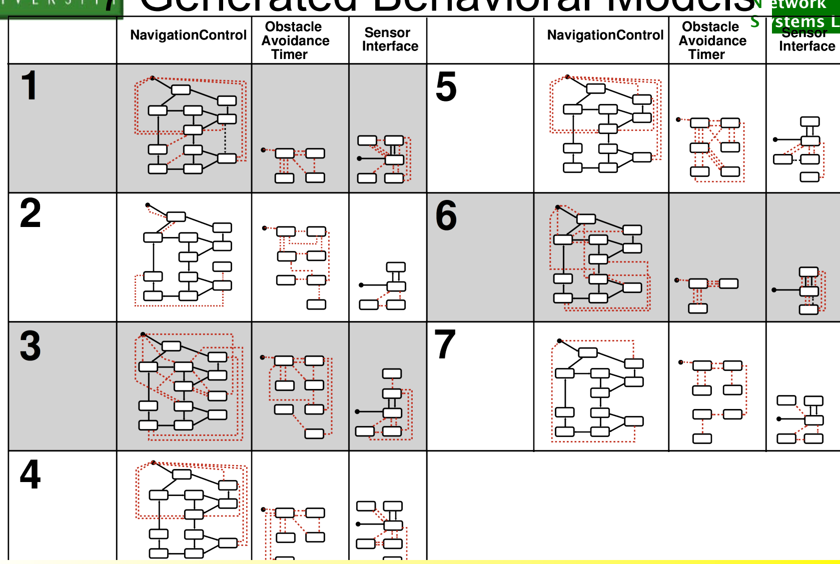
Software Engineering & Network Systems Lab



MICHIGAN STATE UNIVERSITY

7 Generated Behavioral Models

Software Engineering & Network Systems Lab

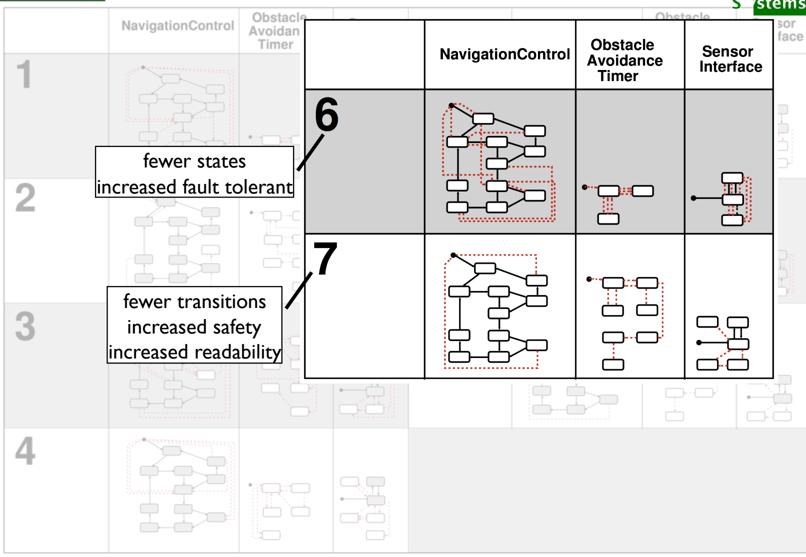


There is structural variation among the generated models…

MICHIGAN STATE UNIVERSITY

7 Generated Behavioral Models

Software Engineering & Network Systems Lab



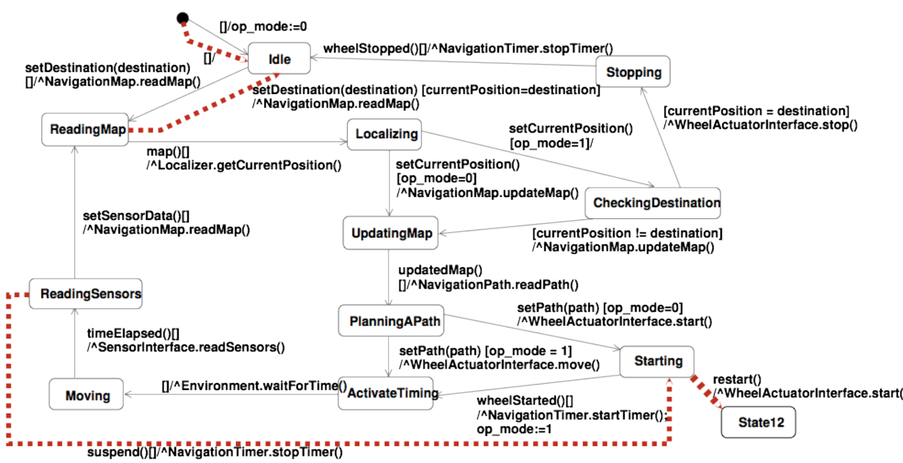
Skip

MICHIGAN STATE UNIVERSITY

NavigationControlState Diagram

Software Engineering & Network Systems Lab

- Extended with 4 additional transitions

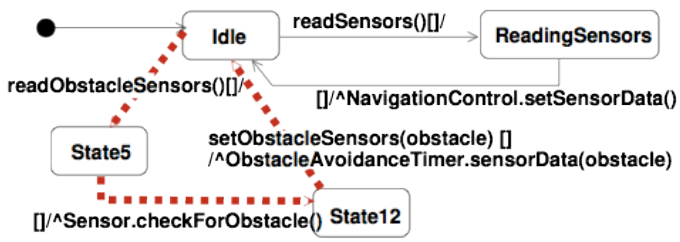


MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

SensorInterface State Diagram

- Extended with 3 additional transitions

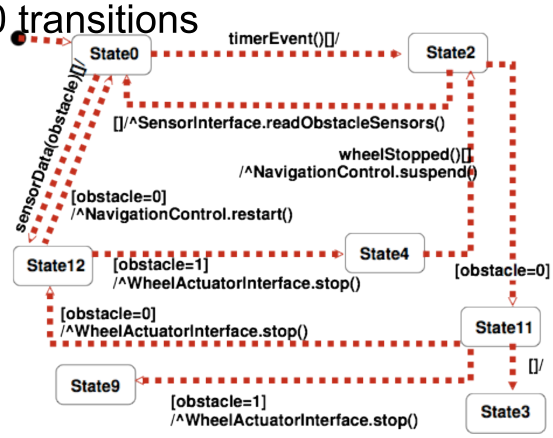


MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

ObstaceAvoidanceTimer State Diagram

- Developed from scratch
- 10 transitions



MICHIGAN STATE UNIVERSITY

Conclusions

Software Engineering & Network Systems Lab

- **Summary:**
  - Harnessing Digital Evolution to automatically generate interesting models of target systems [ICAC08-***Best Paper***]
  - Suite of solutions help address uncertainty in execution environment [MODELS08, ***Distinguished Paper***]
    - Use non-functional criteria to distinguish models
- **Current and Future work** (industrial applications):
  - **Plato-RE:** Use EC for run-time adaptive monitoring [RE@RT2010]
  - **Plato:** Run-time generation of new SW configurations, multi-objective tradeoffs [ICAC09, ***Best Paper***]
  - **Hermes**: Generation of Adaptive Logic [ICAC10]
  - **Loki:** Harness EC for generating interesting environmental conditions (e.g., failures, combinations, attacks, etc.) [ASE11]

MICHIGAN STATE UNIVERSITY

Beyond Orchid

Software Engineering & Network Systems Lab

- **Marple**: Exploring latent properties of software systems
  - Discover implicit (latent) properties [MODELS2010]
  - Detect unwanted properties with Industrial Models (e.g., feature interactions) [MODELS2011, *Best Paper nomination]*
- **RELAX:** a new specification language to explicitly incorporate uncertainty in requirements [RE09, ***One of Best Papers***, REJ10, MODELS09]
- **Medical Applications:**
  - **Polypharmacy:** drug bloat problem for the elderly
- **Automotive Applications:**
  - **Recent:** Onboard electronics (smart cruise, lane keeping, electronic assisted steering)
  - **Current:** Autonomous vehicles (prognostics, self-healing, adaptive, etc.)

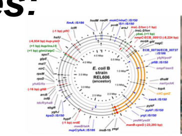
MICHIGAN STATE UNIVERSITY

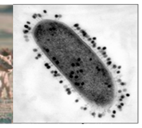
BEACON

Software **E**ngineering & **N**etwork **S**ystems Lab

An NSF Center for the Study of Evolution in Action

- ***Crosscutting Themes:***
  - ★*Biological Evolution*
  - ★*Digital Evolution*
  - ★*Evolutionary Applications*





- **Transformative**: $25 million over 5 years to identify and pursue multi-disciplinary opportunities for research, education, and applications
- **Unifying**: Exchange information, hypotheses, and capabilities:
  - ★3 **thrust groups—evolution of** genomes, behavior, communities
  - ★5 **partners** → chosen for ***their***
    - – crosscutting strengths in evolution
    - – Recruiting Industrial Collaborators for ***Challenge problems***

MICHIGAN STATE UNIVERSITY
UNIVERSITY OF WASHINGTON
University of Idaho
North Carolina A&T State University
THE UNIVERSITY OF TEXAS

BEACON

NSF Cooperative Agreement No. DBI-0939454

MICHIGAN STATE UNIVERSITY

Selected RAPIDware Publications (1)

Software **E**ngineering & **N**etwork **S**ystems Lab

- "Design Patterns for Developing Dynamically Adaptive Systems," (Andres J. Ramirez, Betty H.C. Cheng), in Proceedings of IEEE International Conference on Software Engineering Workshop Software Engineering for Adaptive and Self-Managing Systems (SEAMS), May 2010, Capetown, South Africa, full paper.
- "Applying Design Patterns to an Adaptive News Server," (Andres J. Ramirez and Betty H.C. Cheng) Sixth IEEE International Conference on Autonomic Computing (ICAC09), Barcelona, Spain, June 2009, short paper.•
- "Modular Verification of Dynamically Adaptive Systems" (Ji Zhang, Heather Goldsby, and Betty H.C. Cheng), the Proceedings of Eighth International Conference on Aspect- Oriented Software Development (AOSD09), Charlottesville, Virginia, March 2009.
- "Goal-based Modeling of Dynamically Adaptive System Requirements," (Heather J. Goldsby, Pete Sawyer, Nelly Bencomo, Betty H.C. Cheng, and Danny Hughes), Engi- neering of Computer-Based Systems (ECBS08), Ulster, Northern Ireland, April 2008 (full paper).
- "Specifying Real-time Properties in Autonomic Systems," J. Zhang, Z. Zhou, B.H.C. Cheng, and P.K. McKinley*, Innovations in Systems and Software Engineering*, Springer, vol. 3, number 1, March 2007, pp. 3–16

MICHIGAN STATE UNIVERSITY

Selected RAPIDware Publications (2)

Software Engineering & Network Systems Lab

- "AMOEBA-RT: Run-time Verification of Adaptive Software," (Ji Zhang, Betty H.C. Cheng, and Heather Goldsby) in Proceedings for Workshop on Models at Run-time, selected as a **Best Paper**, Nashville, Tennessee, October 2007.
- "Visualizing the Analysis of Dynamically Adaptive Systems Using i* and DSLs*," (Pete Sawyer, Nelly Bencomo, Danny Hughes, Paul Grace, Heather J. Goldsby, and Betty H.C. Cheng), Proceedings of Second International Workshop on Requirements Engineering Visualization (REV07), New Delhi, India, October 2007.
- "Re-engineering Legacy Systems for Assured Dynamic Adaptation" (Ji Zhang and Betty H.C. Cheng), in Proceedings of IEEE International Conference on Software Engineering Workshop on on Models in Software Engineering (MiSE) May 2007, Minneapolis, MN.
- "Model-Based Development of Dynamically Adaptive Software" (Ji Zhang and Betty H.C. Cheng), in Proceedings of IEEE International Conference on Software Engineering (ICSE06), Shanghai, China, May 2006. (9% acceptance rate) (**Received Distinguished Paper Award**), pp. 371–380.

MICHIGAN STATE UNIVERSITY

Selected RAPIDware Publications (3)

Software Engineering & Network Systems Lab

- "Goal-oriented Specification of Adaptation Requirements in Adaptive Systems" (Gregory Brown, Betty H.C. Cheng, and Ji Zhang), in Proceedings of IEEE ICSE Workshop of Software Engineering of Adaptive and Self-Managing Systems (SEAMS06), May, 2006, pp 23–29.
- "TA-:LTL: Specifying Adaptation Timing Properties in Autonomic Systems" (Z. Zhou, J. Zhang, P. K. McKinley, and B. H. C. Cheng), 3rd IEEE Workshop on Engineering of Autonomic and Autonomous Systems (EASe 2006), Columbia, MD, April 2006.
- " Using Temporal Logic to Specify Adaptive Program Semantics," Ji Zhang and Betty H.C. Cheng, Journal of Systems and Software, Elsevier. Special issue on Architecting Dependable Systems, Eds. R. de Lemos, C. Gacek, and A. Romanovsky, volume 79, issue 10, pp. 1361–1369, October, 2006.
- "Composing Adaptive Software," P. K. McKinley, S. M. Sadjadi, E. P. Kasten, and B. H. C. Cheng, IEEE Computer, vol. 37, no. 7, pp. 56–64, 2004.
- "The Four Levels of Requirements Engineering for and in Dynamic Adaptive Systems" (Daniel M. Berry, Betty H.C. Cheng, and Ji Zhang), Proceedings of 11th International Workshop on Requirements Engineering Foundation for Software Quality (REFSQ), pp. 95-100, June 13-14, 2005, Porto, Portugal.

MICHIGAN STATE UNIVERSITY

Selected RAPIDware Publications (4)

Software Engineering & Network Systems Lab

- "Transparent Shaping of Existing Software to Support Pervasive and Autonomic Computing" (S. Masoud Sadjadi, P.K. McKinley, and B.H.C. Cheng), IEEE ICSE Workshop on Design and Evolution of Autonomic Computing Systems (DEAS), pp. 99–105, St. Louis, Missouri, May 2005. (also accepted for presentation).
- "An Approach to Implementing Dynamic Adaptation in C++" (Scott D. Fleming, B.H.C. Cheng, K. Stirewalt, P.K. McKinley), IEEE ICSE Workshop on Design and Evolution of Autonomic Computing Systems (DEAS), pp. 118–124, St. Louis, Missouri, May 2005.
- "Specifying Adaptation Semantics," (J. Zhang and B.H.C. Cheng), IEEE ICSE Work-shop on Architecting Dependable Systems (WADS), pp. 14–20, St. Louis, Missouri, May 2005.
- "Resource-based Approach to Feature Interaction in Adaptive Software," (J. Bisbal and B.H.C. Cheng), ACM SIGSOFT Workshop on Self-Managing Systems, workshop co- located with ACM SIGSOFT Foundations of Software Engineering (FSE), Newport Beach, CA, October 2004.
- "Enabling Collaborative Adaptation across Legacy Components" (Z. Yang, Z. Zhou, B. H. C. Cheng, and P. K. McKinley), In Proceedings of the Third Workshop on Reflective and Adaptive Middleware (with Middleware'04), pp. 277-282, Toronto, Ontario, Canada, October 2004.

MICHIGAN STATE UNIVERSITY

Selected RAPIDware Publications (5)

Software Engineering & Network Systems Lab

- "TRAP/J: Transparent Generation of Adaptable Java Programs" (S. M. Sadjadi, P. K. McKinley, B. H. C. Cheng, and R. E. K. Stirewalt), In Proceedings of the 2004 International Symposium on Distributed Objects and Applications, Agia Napa, Cyprus, October 2004.
- "Generation of self-optimizing wireless network applications" (S. M. Sadjadi, P. K. McKinley, R. E. K. Stirewalt, and B. H.C. Cheng), Proceedings of of the International Conference on Autonomic Computing (ICAC-04), pages 310-311, New York, NY, May 2004.
- "Adding Safeness to Dynamic Adaptation Techniques," Workshop on Architecture for Dependable Systems (WADS04) associated with IEEE International Conference on Soft- ware Engineering (ICSE04), May 2004, Edinburgh, Scotland, 2004.
- **"An Aspect-Oriented Approach to Dynamic Adaptation" (with Z. Yang, R. E. K. Stire- walt, J. Sowell, S. M. Sadjadi, and P. K. McKinley), Proceedings of the ACM SIGSOFT Workshop on Self-Healing Systems (WOSS02), November 2002.**

MICHIGAN STATE UNIVERSITY

Selected EC Publications (1)

Software **E**ngineering & **N**etwork **S**ystems Lab

- "A Taxonomy of Uncertainty for Dynamically Adaptive Systems" (Andres J. Ramirez and Betty H.C. Cheng), (accepted to appear) in Proceedings of IEEE International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), June, 2012, Zurich, Switzerland.
- "Automatically Exploring How Uncertainty Impacts Behavior of Dynamically Adaptive Systems" (Andres J. Ramirez, Adam C. Jensen, Betty H.C. Cheng and David Knoester,), in Proceedings of 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011), pp. 568–571, November, 2011, Lawrence, Kansas.
- "Applying evolutionary computation to mitigate uncertainty in dynamically-adaptive, high-assurance middleware," (P.K. McKinley, B.H.C. Cheng, A.J. Ramirez, and A.C. Jensen), Journal of Internet Services and Applications, pp. 1–8, 2011, Springer.
- "Plato: A Genetic Algorithm Approach to Run-Time Reconfiguration in Autonomic Computing Systems," Andres Ramirez, David Knoester, Betty H.C. Cheng, and Philip K. McKinley, "Journal of Cluster Computing (Special Issue on Autonomic Computing)," pp. 229–244, vol 14, no. 3, 2011.

MICHIGAN STATE UNIVERSITY

Selected EC Publications (2)

Software **E**ngineering & **N**etwork **S**ystems Lab

- "An Evolutionary Approach to Network Self-Organization and Resilient Data Diffusion" (Andres J. Ramirez and Betty H.C. Cheng), Proceedings Fifth IEEE International Conference on Self-Adaptive and Self-Organizing Systems, pp. 198–207, Ann Arbor, Michigan, USA; October, 2011.
- "Automatic Derivation of Utility Functions for Monitoring Software Requirements" (Andres J. Ramirez and Betty H.C. Cheng), Proceedings of the ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS 2011), pp. 501–516, Wellington, New Zealand, October, 2011.
- "A Toolchain for the Detection of Structural and Behavioral Latent System Properties" (Adam Jensen, Betty H.C. Cheng, Heather J. Goldsby, and Edward Nelson), Proceedings of the ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS 2011), pp. 683–698, Wellington, New Zealand, October, 2011. (**Nominated for Best Paper**).
- "Automatically Discovering Properties that Specify the Latent Behavior of UML Models" (Heather J. Goldsby and Betty H.C. Cheng), Proceedings of the ACM/IEEE Inter- national Conference on Model Driven Engineering Languages and Systems (MODELS 2010), pp. 316–330, Oslo, Norway, October 2010.

MICHIGAN STATE UNIVERSITY

Selected EC Publications (3)

Software Engineering & Network Systems Lab

- "Adaptive Monitoring of Software Requirements," (Andres J. Ramirez, Betty H.C. Cheng, and Philip K. McKinley), In the Proceedings of First International Workshop on Requirements Engineering (RRT10), Sydney, Australia. September, 2010.
- "Automatically Generating Adaptive Logic to Balance Non-functional Tradeoffs During Reconfiguration" (Andres J. Ramirez, Betty H.C. Cheng, Philip K. McKinley, and Benjamin E. Beckmann), In the Proceedings of the 7th International Conference on Autonomic Computing (ICAC-10), Washington, DC, USA, pp. 225-234.
- "Evolving Models at Run Time to Address Functional and Non-Functional Adaptation Requirements," (Andres J. Ramirez and Betty H.C. Cheng), In the Proceedings of the Fourth International Workshop on Models at Run Time (MaRT09), Denver, Colorado, USA, October, 2009.
- "Applying Genetic Algorithms to Decision Making in Autonomic Computing Systems," (Andres J. Ramirez, Betty H.C. Cheng, David Knoester, and Philip K. McKinley) Sixth IEEE International Conference on Autonomic Computing (ICAC09), pp. 97– 106, Barcelona, Spain, June 2009, (received **Best Paper Award**). (Full paper, 16% acceptance rate).
- "Automatically Generating Behavioral Models of Adaptive Systems to Address Uncertainty" (Heather Goldsby and Betty H.C. Cheng), the Proceedings of the ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MoDELS 2008), Toulouse, France, October 2008 (Selected as one of the **Best Papers**).

MICHIGAN STATE UNIVERSITY

Selected EC Publications (4)

Software Engineering & Network Systems Lab

- "Harnessing Digital Evolution," P. McKinley, B.H.C. Cheng, C. Ofria, D. Knoester, Be. Beckmann, and H. Goldsby, IEEE Computer, volume 41, number 1, January 2008, pp. 54–63.

- **"Digitally Evolving Models for Dynamically Adaptive Systems" (Heather J. Goldsby, David B. Knoester, Betty H.C. Cheng, Philip K. McKinley, and Charles A. Ofria), in Proceedings of IEEE International Conference on Software Engineering Workshop Software Engineering for Adaptive and Self-Managing Systems (SEAMS), May 2007, Minneapolis, MN.**

MICHIGAN STATE UNIVERSITY

Software Engineering & Network Systems Lab

Model-Based Approach to High-Assurance Dynamically Adaptive Systems

Acknowledgements:

Research Team: Heather Goldsby, Philip McKinley, Dave Knoester, Charles Ofria, Xiaobo Tan, Andres Ramirez, Chad Byers, Adam Jensen, Ben Beckmann, Erik Fredericks, and many others from SENS Lab, DevoLab, and BEACON Center

Current Funding:

NSF Grants: CCF-0820220, CNS-0751155, IIP-0700329,

CNS-0854931, Army Research Grant W911NF-08-1-0495, Ford