MICHIGAN STATE UNIVERSITY

An Aspect-Oriented Approach to Dynamic Adaptation

Betty H.C. Cheng
Software Engineering and Network Systems Lab
Department of Computer Science and Engineering
Michigan State University
http://www.cse.msu.edu/SENS

Co-Authors: Z. Yang, K. Stirewalt, M. Sadjadis, J. Sowell, and P. McKinley

This work is supported in part by grants from NSF (EIA-0000433, EIA-0130724, CCR-9901017, CCR-9984726), ONR (N00014-01-1-0744)

SENS

Observations from yesterday…

- "One size does not fit all"
  - Different types of adaptation for different needs
- Adaptation may occur at different levels of abstraction:
  - "Architecture" vs "Infrastructure"
- *Design* for adaptation vs *retrofitting* legacy system
- *Internal* vs *external* monitoring for adaptation

MICHIGAN STATE UNIVERSITY

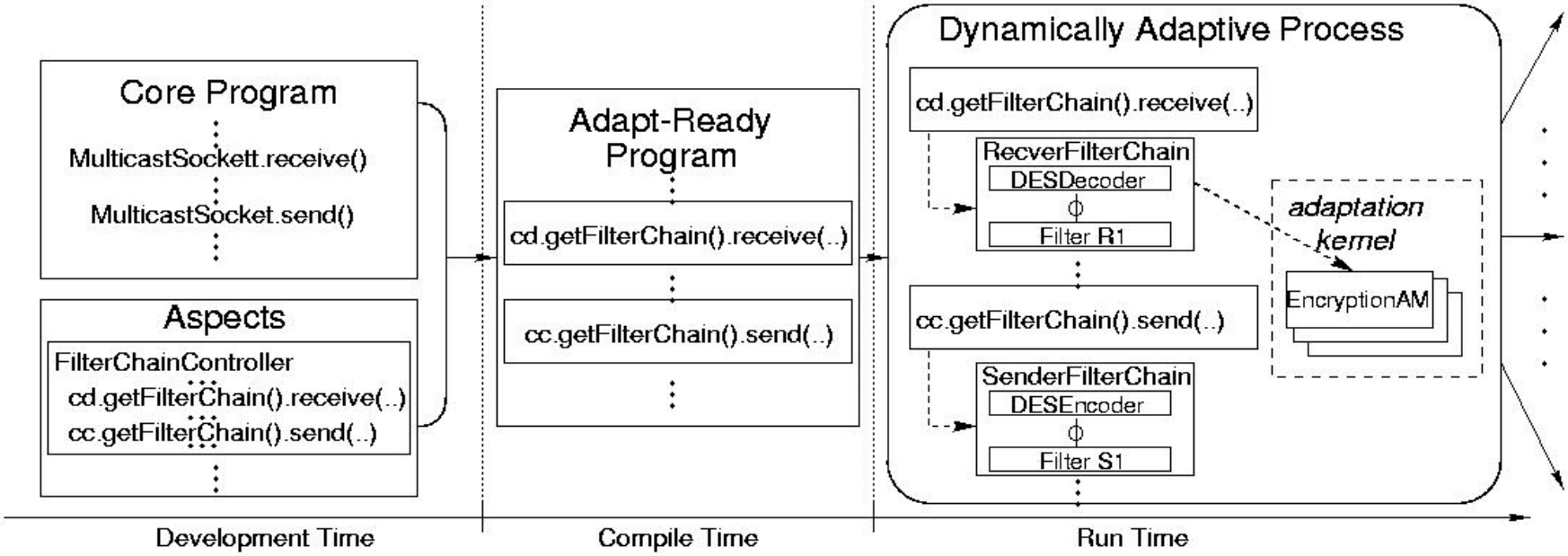
Highlights of Our Approach

- Infrastructure Abstractions:
  - Through which applications interact with environment
  - Example: multicast sockets
- Objective: Self-healing abstractions
  - Extend abstractions to make them adaptable to environmental changes.
  - Example: Multicast socket extension pipes data through filters.
- Observation:
  - Self-healing version may require changes to API
  - Thus potentially affecting compatibility with client code
- AOP enables non-invasive migration to new abstractions
  - Modify call sites to use new abstractions
  - Maintain traceability to original program

SENS

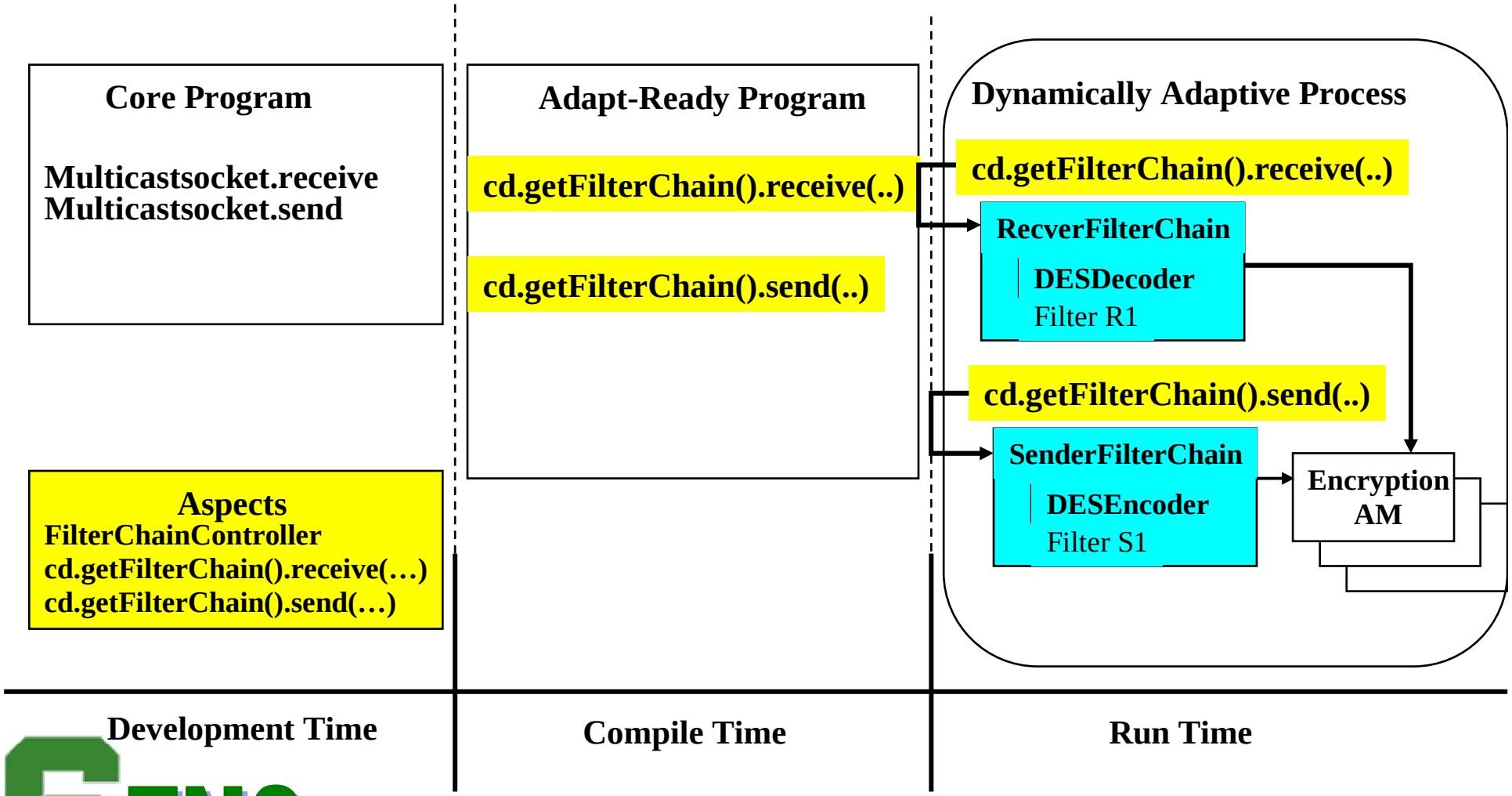
MICHIGAN STATE UNIVERSITY

The Big Picture



SENS

The Big Picture



MICHIGAN STATE UNIVERSITY

Target Applications

- What applications are we targeting?
  - General: online, distributed, collaborative applications
  - Specific example in paper: Java-based online conferencing
- Self-healing scenarios:
  - Intrusion scenarios:
    - **Detect 1:** participant from unknown IP address joins in conferencing
    - **Correct 1:** insert encryption/decryption software to secure all transmissions
    - **Detect 2:** malicious user sends flooding messages
    - **Correct 2:** insert a piece of new code that filters flooding messages
  - QoS scenario:
    - **Detect 3:** network becomes overly crowded, causing many messages to be lost
    - **Correct 3:** insert FEC (forward error correction) facilities and tune FEC parameters

SENS

MICHIGAN STATE UNIVERSITY

Meaning of Self-Healing

- What does "self-healing" mean to us?
  - When an application encounters ***changes in the environment*** that lead to undesirable behavior (service degradation, security violation, etc.),
  - It can adapt the application to account for the changes in the environment by ***introducing new code*** or removing (previously inserted) code ***at runtime***

SENS

MICHIGAN STATE UNIVERSITY

Scope of Self-Healing

- What part of the self-healing problem are we dealing with?
  - **Detection:** determine when changes in the environment result in undesirable behavior
  - **Correction:** adapt application at runtime to respond to environmental conditions
  - **Programming language**: use existing language features to support the incorporation of self-healing abstractions into existing code.

SENS

MICHIGAN STATE UNIVERSITY

Relevant Papers

- Eric P. Kasten, Philips K. McKinley, Seyed Masoud Sadjadi, and R. E. Kurt Stirewalt. Separating introspection and intercession in metamorphic distributed systems. In Proceedings of the IEEE Workshop on Aspect-Oriented Programming for Distributed Computing (with ICDCS'02), Vienna, Austria, July 2002.

- Gordon Blair, Geoff Coulson, and Nigel Davies. Adaptive middleware for mobile multimedia applications. In Proceedings of the 8th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV), pages 259-273, 1997.

- Ian Welch and Robert Stroud. Dynamic adaptation of the security properties of applications and components. In ECOOP Workshop on Distributed Object Security, Brussels, Belgium, 1998.

- Israel Ben-Shaul, Ophir Holder, and Boris Lavva. Dynamic adaptation and deployment of distributed components in Hadas. IEEE Transactions on Software Engineering, 27(9):769-787, September 2001.

SENS

MICHIGAN STATE UNIVERSITY

Details of Example Application

Rule of Adaptation:

<Detect: out of time,

Correct: skip remaining slides>

SENS

MICHIGAN STATE UNIVERSITY

Adaptive Conference Application

- Online conferencing application
- Dynamic Adaptation Phases
  - **Phase I:** making it adapt-ready
    - Define adaptation points
    - Insert adaptation infrastructure
  - **Phase II**: runtime adaptation
    - Check conditions from rule base
    - Dynamically load code if conditions are satisfied

SENS

MICHIGAN STATE UNIVERSITY

Phase I

- Define Adaptation Points

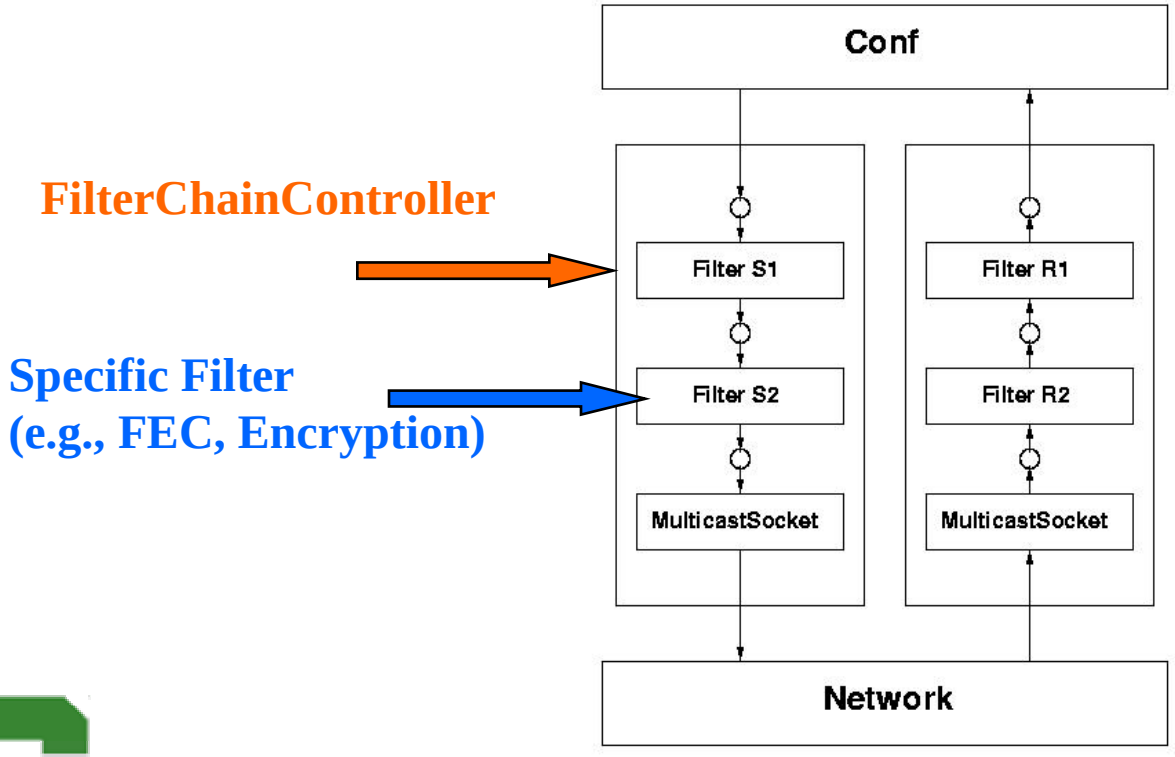
```
pointcut receive(MulticastSocket ms,
                                 DatagramPacket dp):
          target(ms)
        && args(dp)
        && call(public *

*..MulticastSocket.receive(DatagramPacket)
   );
```

SENS

Phase I – cont'd

- Insert Adaptation Infrastructure



MICHIGAN STATE UNIVERSITY

Phase II

- Runtime adaptation
- Check rule base:
  - Comprising <condition,action> pairs
  - An Example Rule

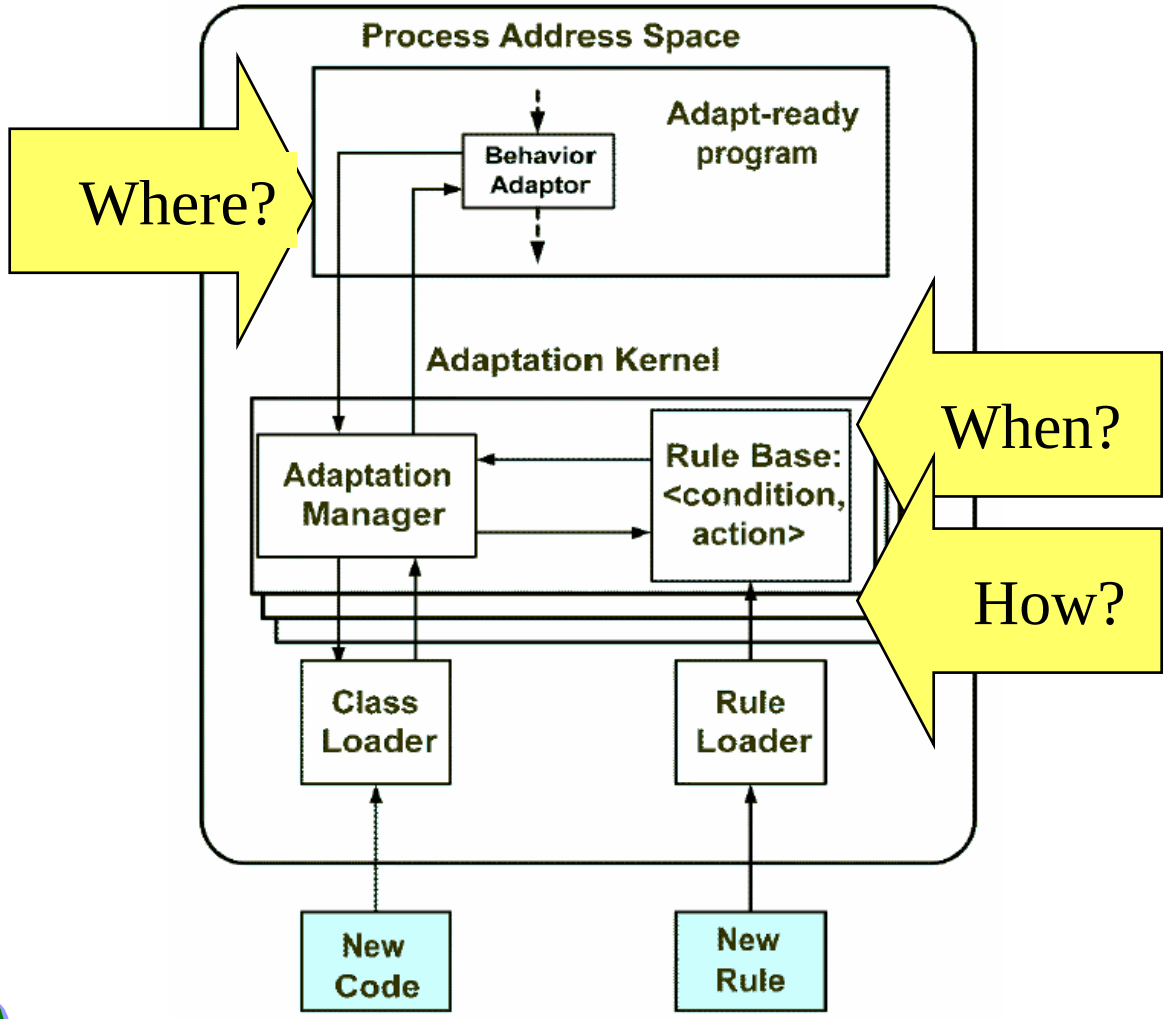
```
DESController=                          % name
ippattern=35.9.20.19                    % condition
&action=edu.msu.cse.sens.conf.          % action
        adapt.security.crypto.
        InsertDESFilters
```

- **Load new code as dictated by action**

SENS

MICHIGAN STATE UNIVERSITY

AOP-Based Dynamic Adaptation



SENS

MICHIGAN STATE UNIVERSITY

Conclusions and Future Work

- The AOP-based dynamic adaptation
  - uses rules to direct dynamic adaptation
  - fully separates application code from dynamic adaptation concern
- Future Work
  - A generic way to define adaptation points
  - Other rule-based dynamic adaptation frameworks besides the AOP-based approach

SENS

MICHIGAN STATE UNIVERSITY

References

[1] Z. Yang, B. H. C. Cheng, K. Stirewalt, M. Sadjadis, J. Sowell, and P. McKinley, "***An aspect-oriented approach to dynamic adaptation,***" in Proceedings of the ACM SIGSOFT Workshop on Self-Healing Systems (WOSS02), Nov. 2002.

[2] G. Kiczales, J. Lamping, A. Menhdhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin, "***Aspect-oriented programming***," in ECOOP '97 Object-Oriented Programming 11th European Conference," Finland (M. Aksit and S. Matsuoka, eds.), vol. 1241, pp. 220-242, New York, NY: Springer-Verlag, 1997.

SENS