

# Cataloging and Detecting Architectural Bad Smells

Joshua Garcia, Daniel Popescu, and  
Nenad Medvidovic, University of  
Southern California

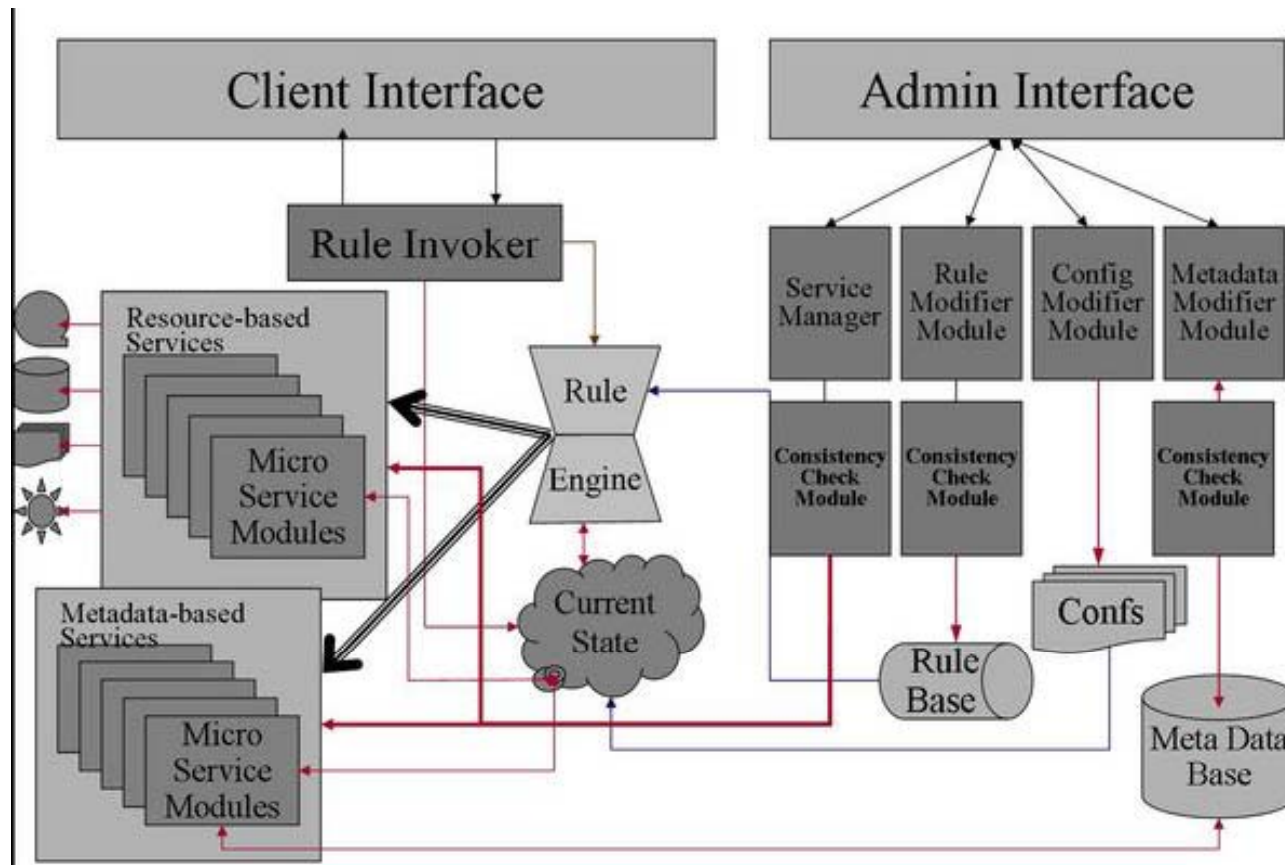
Yuanfang Cai, Drexel University



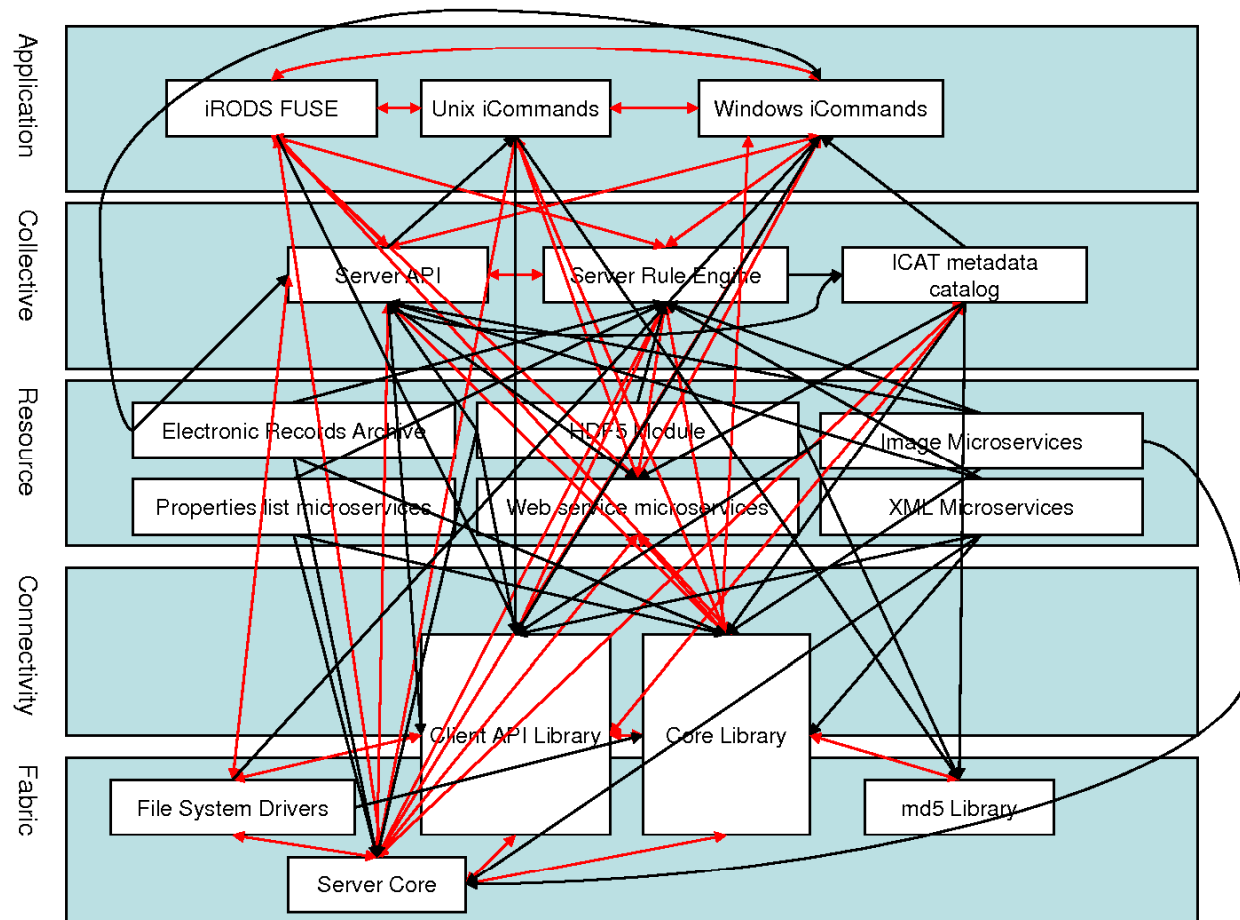
# Motivation – Refactoring Architecture

- **Successful systems are maintained over multiple years**
- **System's Life-Cycle Properties worsen over time**
  - Understandability
  - Testability
  - Extensibility
  - Reusability
- **Restructuring/Refactoring helps to improve life-cycle properties**
  - Code Smells
- **When and where to refactor a software system's architecture?**

# Motivation – iRODS - Prescriptive



# Motivation – iRODS - Descriptive

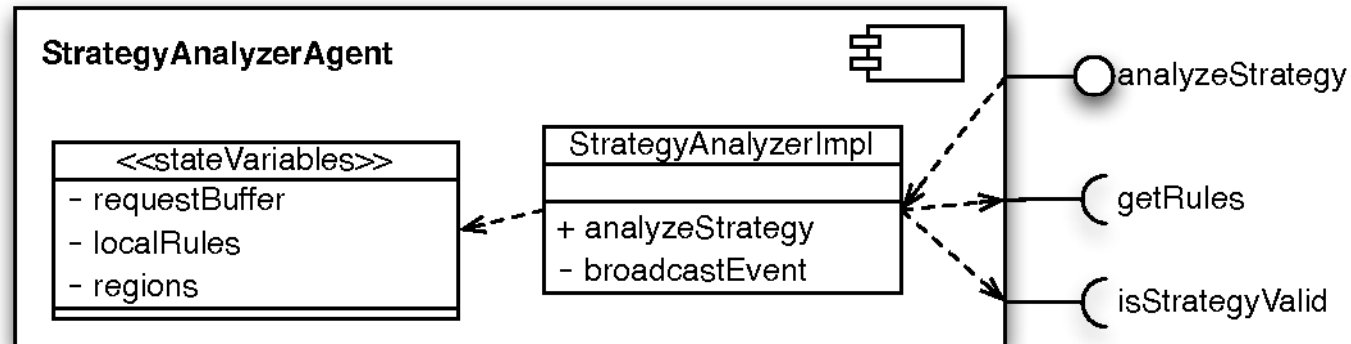


# Contribution and Goals



- **Categorization of Architectural Smells**
  - Components, Connectors, Interfaces, Data Elements, Concerns
  - Separation of Concerns
  - Coupling and Cohesion
- **Novel Architectural Recovery Technique**
  - Identification of Elements
  - Concern Meta-Classification
- **Novel Architecture Representation**
  - Extended Augmented Constraint Network
  - Design Structure Matrix
- **Architectural Smell Detection**

# Connector Envy - Example



- **Component exhibiting interaction-related functionality that should be delegated to connector**
  - Reusability, understandability, testability
- **StrategyAnalyzerAgent from Emergency Response System**

# Architectural Recovery for Smells



- **Component identification**
  - Hierarchical clustering
  - Concerns through topic modeling
    - SAA – “strategy,” “rule,” “region”
- **Connector identification**
  - Pattern matching, Supervised Learning
- **Interface and Data Element Identification**
- **Concern Meta-classification**
  - Application-specific or connector-oriented concern

# Novel Architecture Representation



- **Extended Augmented Constraint Network**
  - Uniform, formal way of capturing of architectural decisions
  - Constraint network, design rule, cluster set, concerns from topic models

# Design Structure Matrix of ERS

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1 Personnel Resources	1																										
2 UI elements		2																									
3 Event and Message Management			3																								
4 Agent Rendering				4																							
5 Prism Architecture Object Management					5																						
6 Weather						6																					
7 Commander and Agents							7																				
8 Shared Data Structure Elements								8																			
9 Usage of Shared Data Structures									9																		
10 Main UI Frame										10																	
11 DataInterface											11	x															
12 ComponentInterface											x	12	x														
13 ConnectorInterface													13														
14 Component_RenderingAgent		0.56		0.24						0.19	x	x		14													
15 Component_SimulationAgent	0.36		0.41				0.09	0.06	0.08		x	x			15					x							
16 Component_ResourceManager	0.48		0.47								x	x				16											
17 Component_SAKBUI			0.31				0.19	0.29	0.21		x	x					17										
18 Component_StrategyAnalyzer			0.33				0.17	0.31	0.19		x	x	x					18									
19 Component_DeploymentAdvisor			0.14						0.32		x	x							19	x							
20 Component_ResourceMonitor			0.49	0.13		0.23			0.14		x	x	x							20							
21 Component_map			0.43			0.07			0.10		x	x									21						
22 Component_Repository			0.30			0.17			0.06		x	x										22					
23 Component_Weather			0.30			0.20			0.19		x	x											23				
24 Component_WeatherAnalyzer			0.27						0.23		x	x												24			
25 Component_StrategyAnalysisKB			0.54				0.33		0.13		x	x													25		
26 Component_Clock		0.20	0.25						0.30		x	x														26	
27 Commander					0.49	1.00					x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	17

# Thank You



- **Thank You**

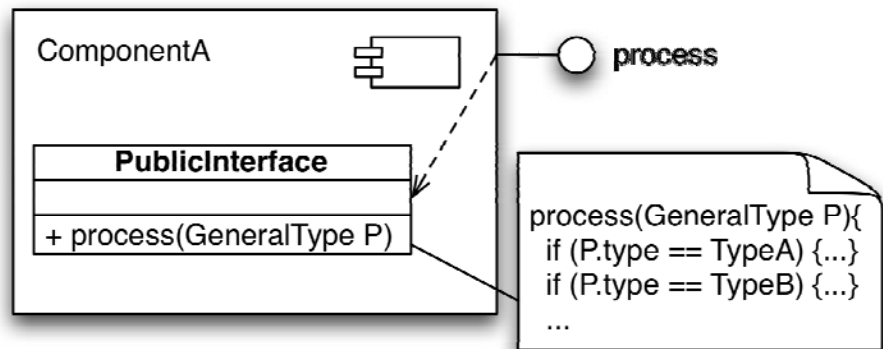
# Smells of Different Granularities



- **Code smell**
  - Code smells are *implementation* structures that negatively affect system lifecycle properties
- **Defined in terms of *implementation-level* constructs**
  - Classes
  - Methods
  - Statements
- **Examples**
  - Long parameter list
  - Large methods
- **Code smells do not necessarily address *architectural decisions***
- **Architectural smell**
  - A commonly used *architectural* decision that negatively impacts lifecycle properties
- **Possible Causes**
  - Applying a design solution in an inappropriate context
  - Mixing design fragments that have undesirable emergent behaviors
- **Architectural Refactoring – The remedy**
  - Altering the internal structure of the system
  - Altering the behaviors of internal system elements
  - Avoid changing external system behavior

# Ambiguous Interfaces – Description

- An *Ambiguous Interface* offers only one public interface
- Internally dispatches to multiple services
- Appears especially in event-based publish-subscribe systems
  - Example: JMS
- User has to inspect the component's implementation before knowing about its offered services
- Negatively affects
  - Analyzability
  - Understandability



# Connector Envy – Description

- **Connector roles**

- Communication
- Coordination
- Conversion
- Facilitation

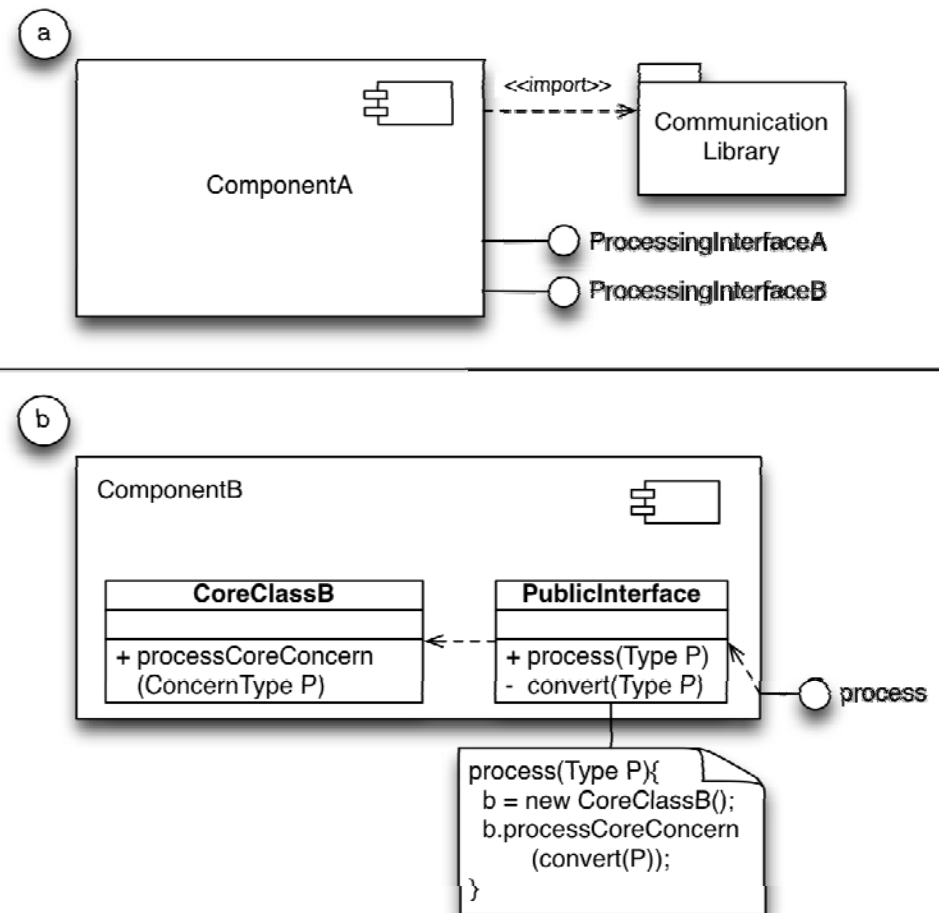
- **Components with *Connector Envy* encompass extensive interaction-related functionality**

- Example: Gridfarm Filesystem Daemon

- **Violates separation of concerns**

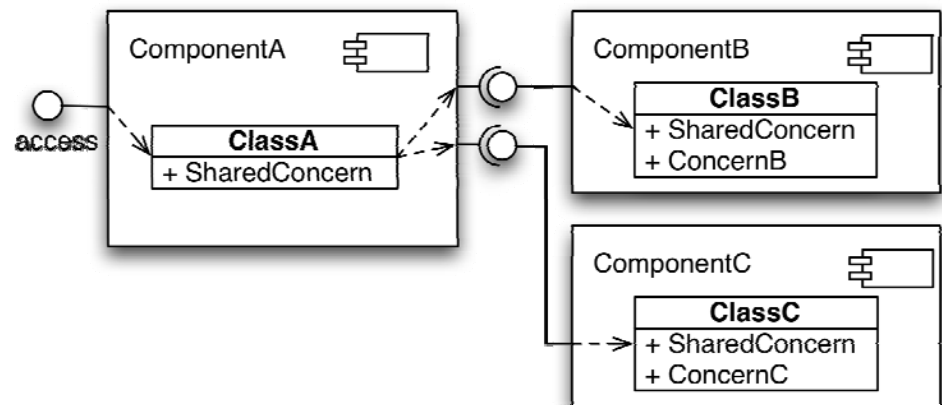
- **Negatively affects**

- Reusability
- Understandability
- Testability



# Scattered Functionality – Description

- Multiple components are responsible for realizing the same high-level concern
- Some of those components are responsible for orthogonal concerns
  - Example: Linux's Status Reporting
- Violates the principle of separation of concerns *twice*
- Negatively affects
  - Reusability
  - Understandability
  - Testability



# Extraneous Connector - Description

- **Two connectors of different types are used to link a pair of component**
  - Example: Events vs. Procedure Calls
  - Example system: Example: Old MIDAS version
- **Benefits of each connector type may cancel each other out**
- **This example negatively impacts**
  - Understandability
  - Reusability
  - Adaptability

