Department of
Information Engineering
and Computer Science
DISI
UNIVERSITY
OF TRENTO - Italy

From Requirements to Architecture for Adaptive Software Systems

Re-Seminar

November 2nd 2012

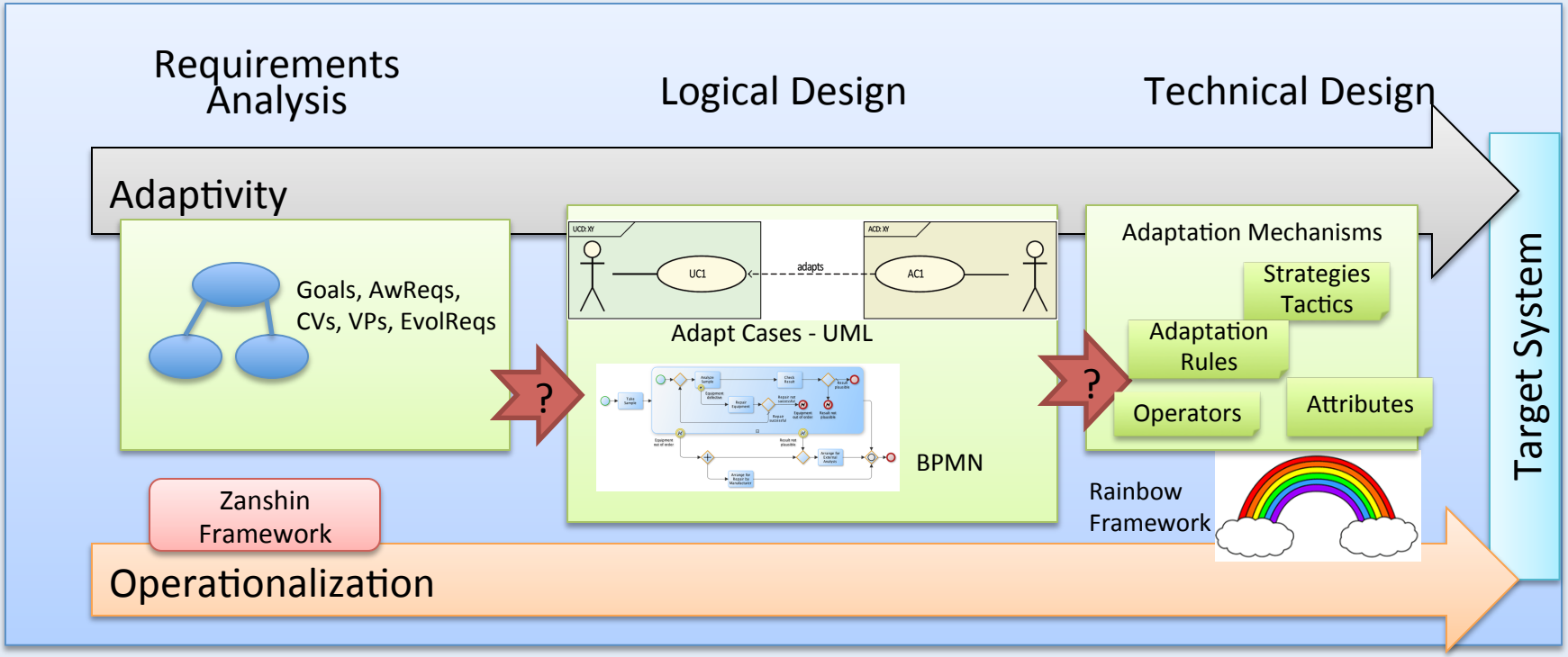
Agenda

- Introduction
- The Requirements - Architecture Gap
- Motivating Scenario
- Requirements vs Design
- Conclusions
- Future Work

Introduction

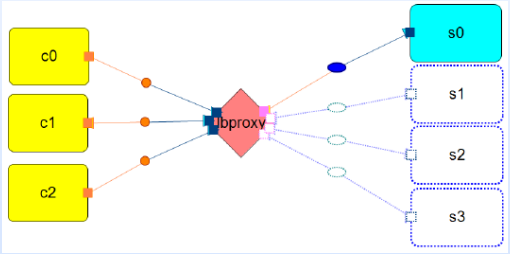
- **Fact:** Adaptation is a process that takes place when changes happen.
- **Fact:** When adaptation takes place changes are needed.
- RQ: How do we model and govern these changes ?
- A: Provide a top-down development approach that captures the adaptation concept starting from requirements level, then design and finally architecture.

Bridging The Gap



Motivating Scenario

Znn.com: An news portal with multimedia content.

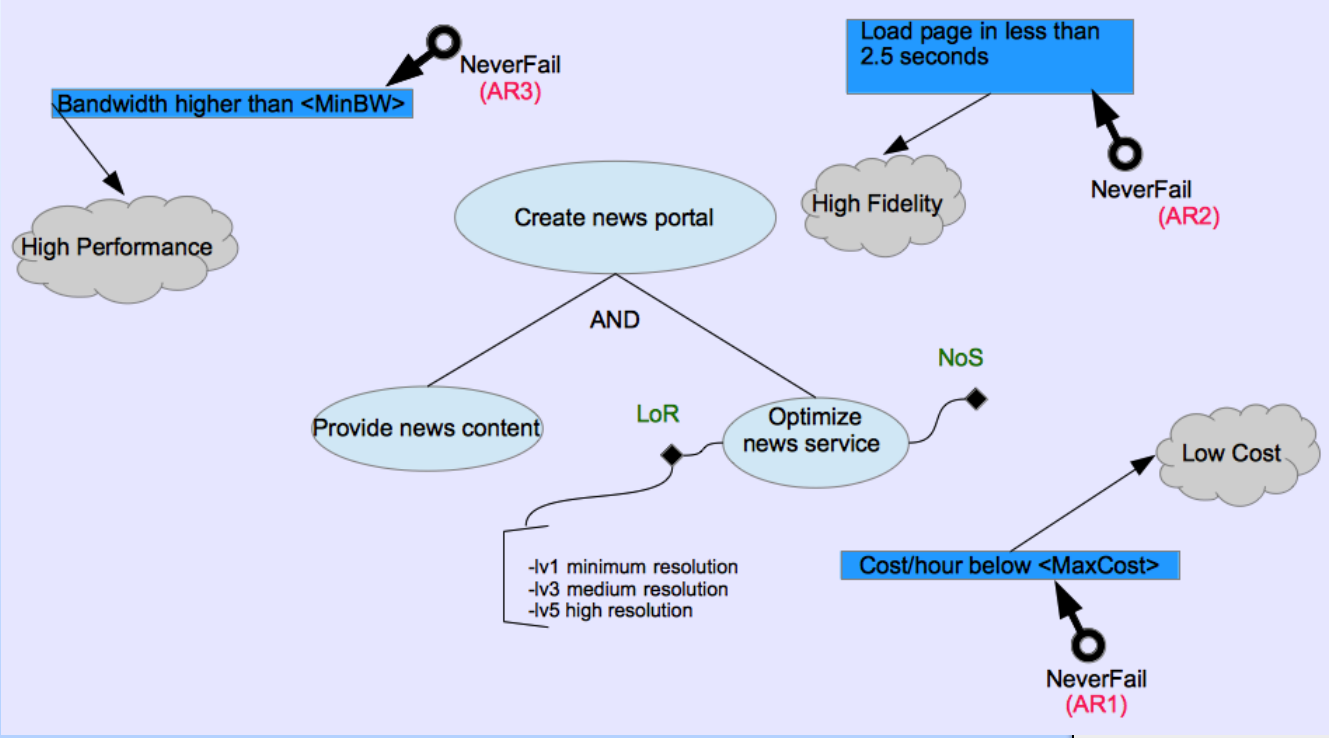


When the load is high there are 2 possible actions:

1. Add more servers
2. Switch to textual mode

When balance is achieved reverse adaptation processes take place to reduce the operational cost and increase the fidelity.

Requirements Approach (1)



Requirements Approach (2)

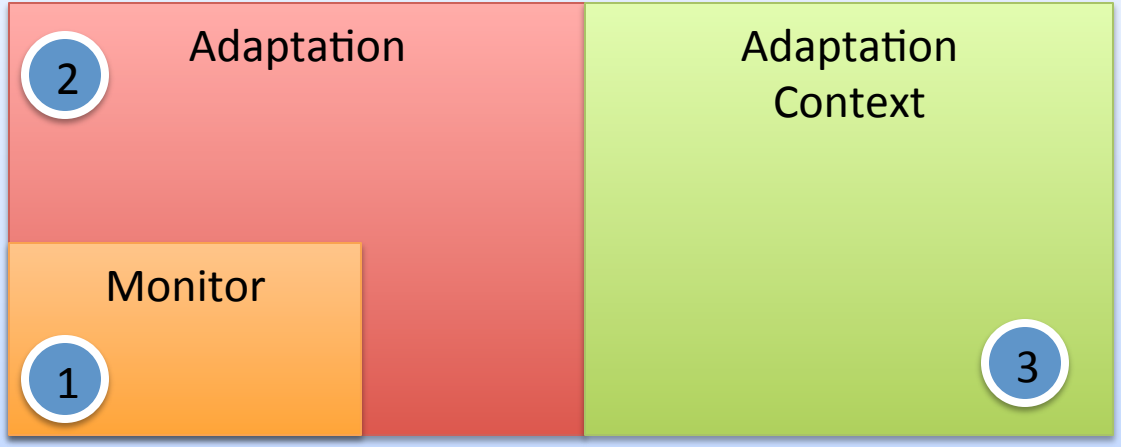
Differential Equations

- Δ (AR1/NoS)[0,maxSrv] < 0
- Δ (AR3/NoS)[0,maxSrv] > 0
- Δ(AR2/LoR)[lv1➔ lv2 ➔ lv3] < 0
- Δ (AR3/LoR)[lv1➔ Lv2 ➔ lv3] < 0

Control Variables:

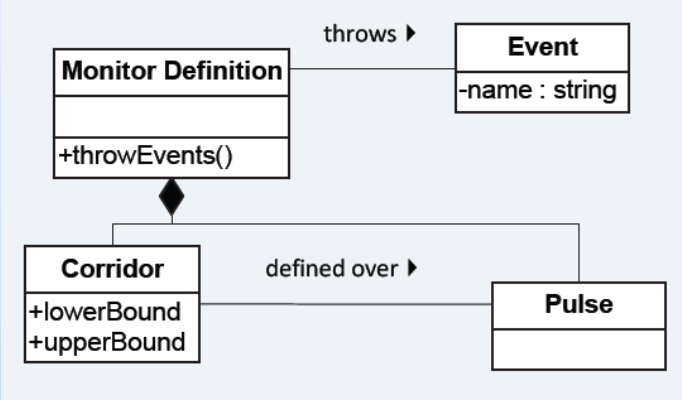
- NoS: Number of Servers
- LoR: Level of Resolution

Adapt Cases Baseline

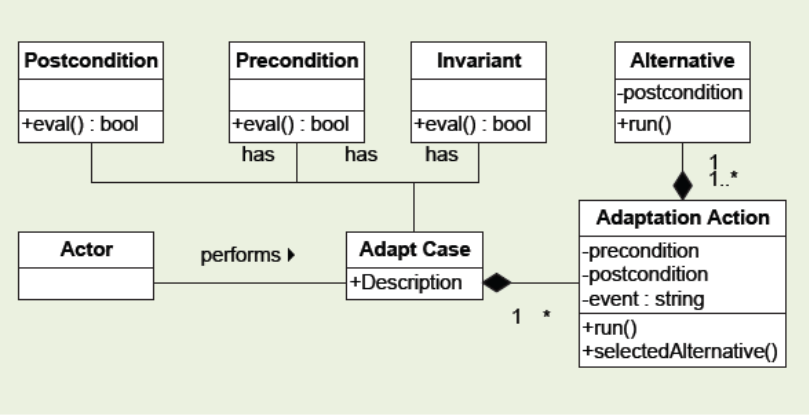


Adapt Cases - Monitor

- **Monitor Definition:** Represents Monitor Concepts.
- **Corridor:** Defines valid values of adaptation context elements.
- **Pulse:** Represents value or status of monitored element; may aggregate values.
- **Event:** Indicates change in adaptation context



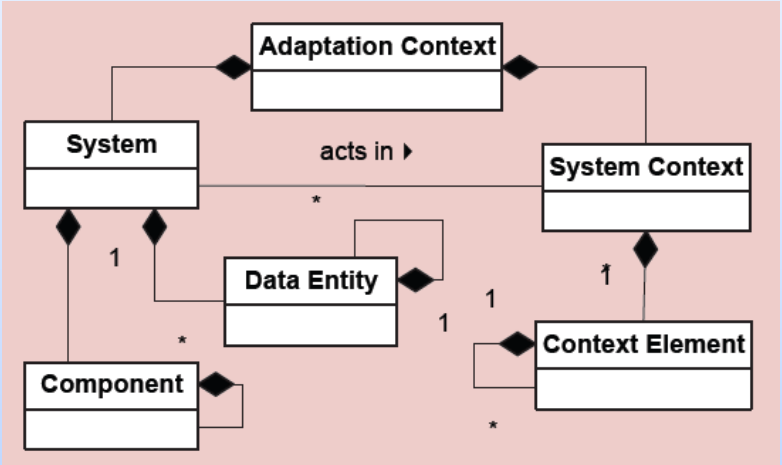
Adapt Cases – Adaptation Action



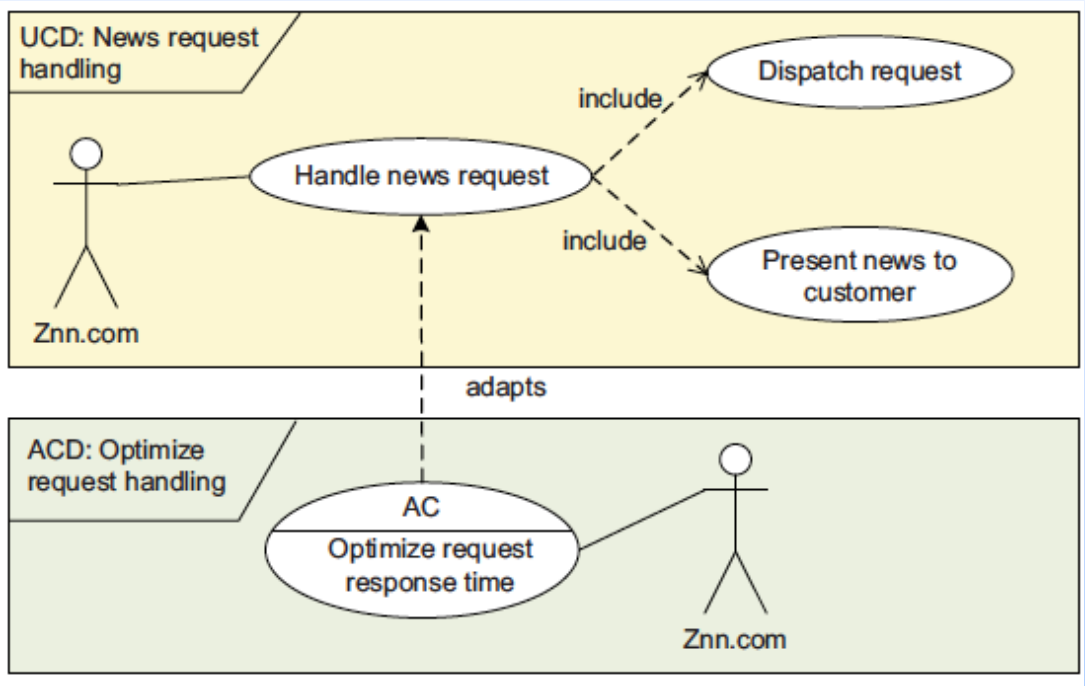
- **Actor:** Performs or is involved in the Adapt Case.
- **Conditions:** Defines Pre- and Postconditions and Invariant.
- **Adaptation Action:** Reacts to event and defines alternative actions.
- **Alternative:** Description of concrete adaptation.

Adapt Cases – Adaptation Context

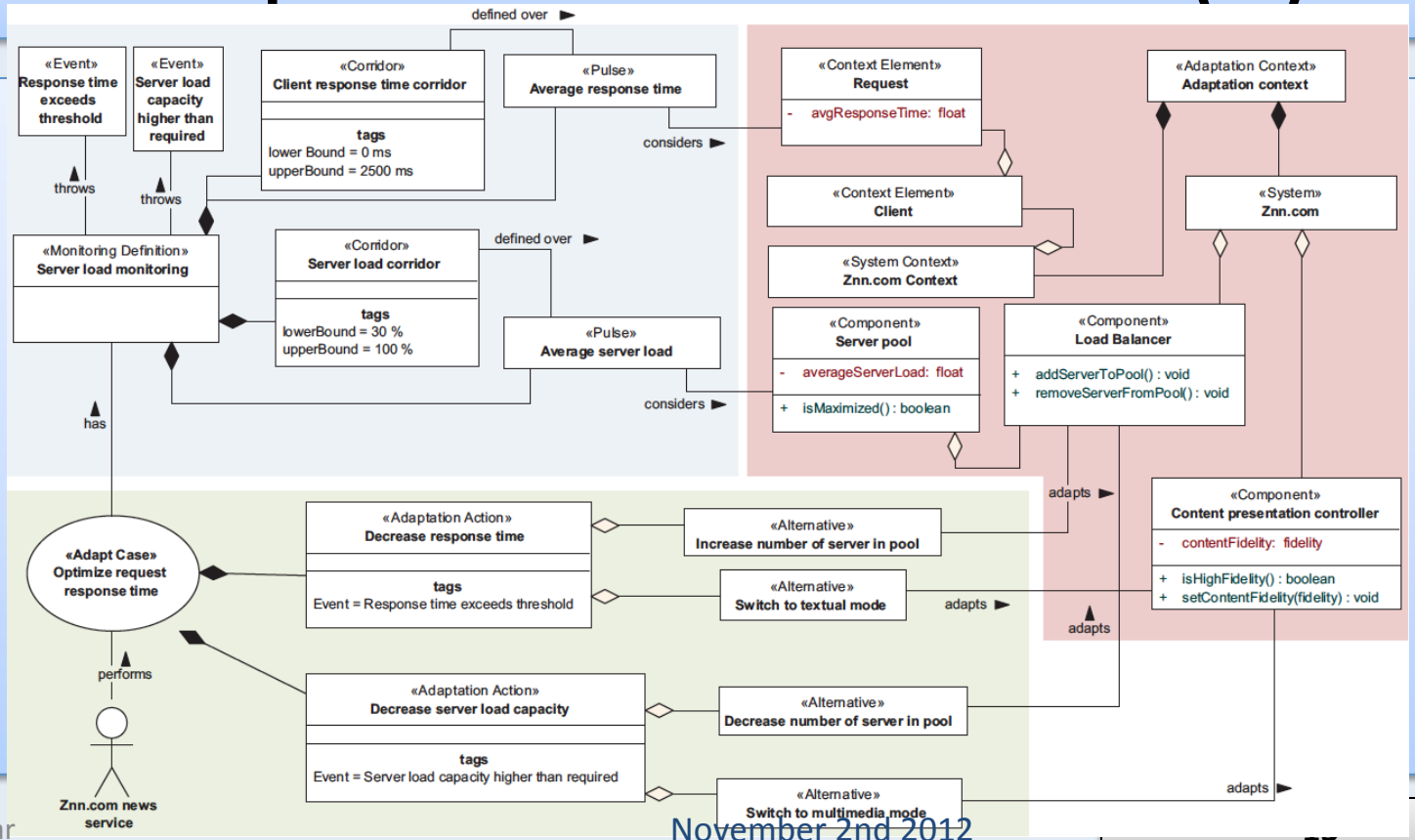
- **Adaptation Context:** Overall context, the system is located in.
- **System:** View on the system that is adapted by the Adapt Case.
- **Component**: Represents logical system component.
- **Data Entity:** Describes data stored by the system.
- **System Context:** View on the context, the system is acting in.
- **System Context Element:** Any kind of element in the system context.



Adapt Cases – Znn.com (1)



Adapt Cases – Znn.com (2)



Adapt Cases – Znn.com (3)

```
def IncreaseNumberOfServerInPool()
 component->select(cmp|cmp.type =
  "LoadBalancer").addServerToPool()

def SwitchToTextualMode()
  component->select(cmp|cmp.type =
      "ContentPresentationController")
   .setContentFidelity(text);

def DecreaseResponseTime.run()
  if ServerPool.isMaximized() then
    SwitchToTextualMode.run()
  else
    IncreaseNumberOfServerInPool.run()
  endif
```

```
def DecreaseNumberOfServerInPool()
 component->select(cmp|cmp.type =
  "LoadBalancer").removeServerFromPool()

def SwitchToMultimediaMode()
  component->select(cmp|cmp.type =
      "ContentPresentationController")
   .setContentFidelity(multimedia);

def DecreaseServerLoadCapacity.run()
  if ContentPresentationController.isHighFidelity
      () then
    DecreaseNumberOfServerInPool.run()
  else
    SwitchToMultimediaMode.run()
  endif
```

Comparison Points

Requirements Approach

1. Follows the MAPE model
2. Monitoring the Awareness Requirements
3. Awareness Requirements operationalized with OCL
4. Variability is expressed with Variation Points
5. Control Variables that compose the factors to be adjusted. (can be added to the components)
6. Differential equations between input and output

Adapt Cases

1. Follows the MAPE-K model (adds context)
2. Monitoring the Pulses
3. Adaptation actions/conditions are expressed in OCL
4. Variability is expressed with Alternatives
5. Express technical details about the system (e.g. components)

Conclusions

- Links between Requirements and Design exist
- Both approaches are based on control feedback loops
- Design gives a more detailed view of the final system
- Adapt Cases model the explicitly the adaptation process by extending an already widely accepted language (UML)

Future Work

- Add Evolution Requirements to the Design
- Use BPMN/BPEL to model the behavior aspect of the adaptation process
- Use TROPOS to capture also social aspects of the design instead of plain goal models
- Link Adapt Cases with Rainbow
- Final Step: Develop a framework that unifies Requirements, Design and Architecture


QUESTIONS?

THANK YOU!