



Università degli Studi di Trento
Facoltà di Scienze Matematiche, Fisiche e Naturali
Dipartimento di Ingegneria e Scienza
dell'Informazione

(Requirement) Evolution Requirements for Adaptive Systems

Vítor E. Silva Souza

vitorsouza@disi.unitn.it
<http://disi.unitn.it/~vitorsouza>

License to use, adapt and distribute

This material is available for any kind of use and can be derived and/or redistributed, as long as it uses an equivalent license and attributes credit to original authors.




Attribution-Share Alike 3.0
Unported

<http://creativecommons.org/licenses/by-sa/3.0/>

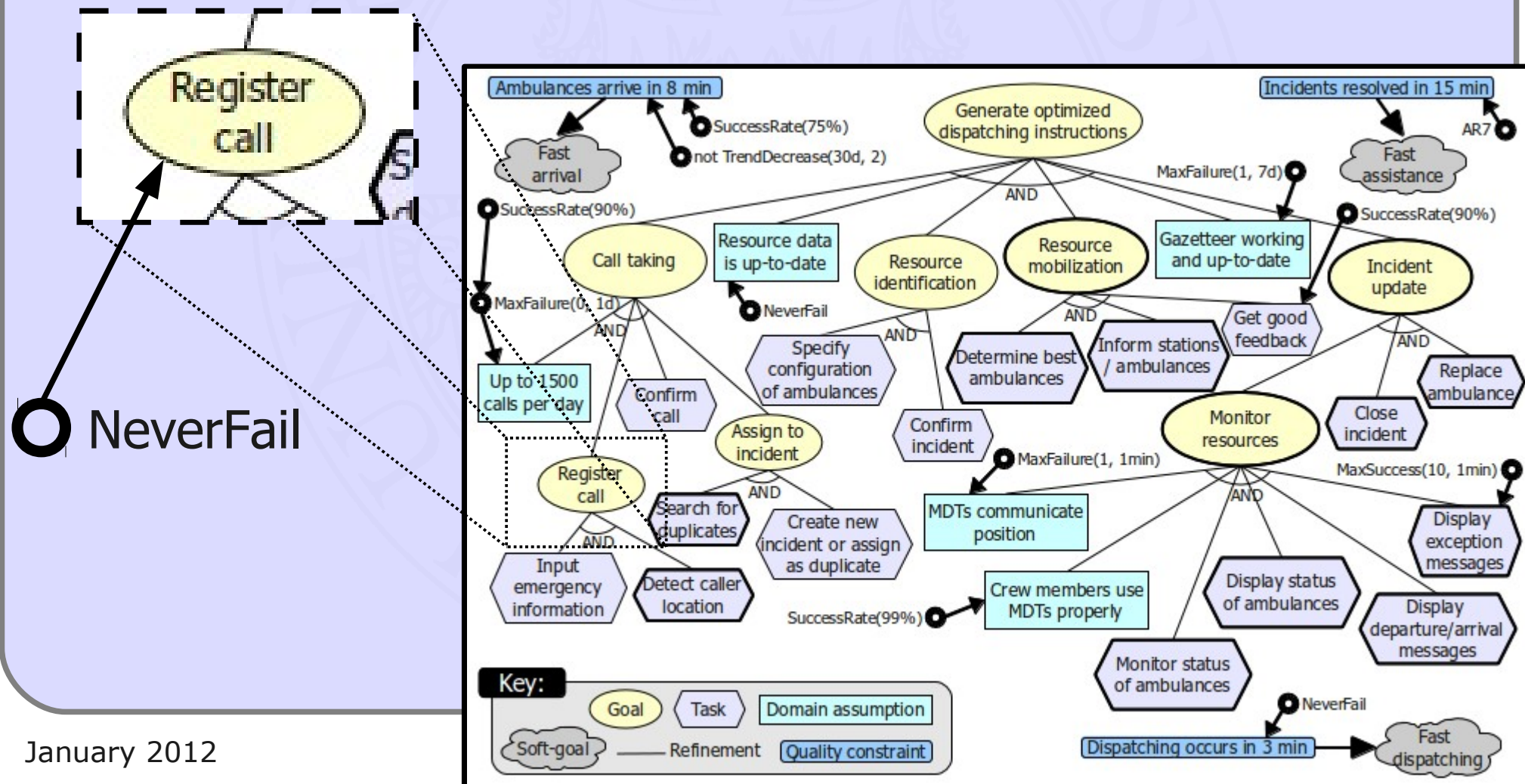
You are free to copy, distribute, transmit and adapt this work under the following conditions: (a) You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work); (b) If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.

What I will present today...

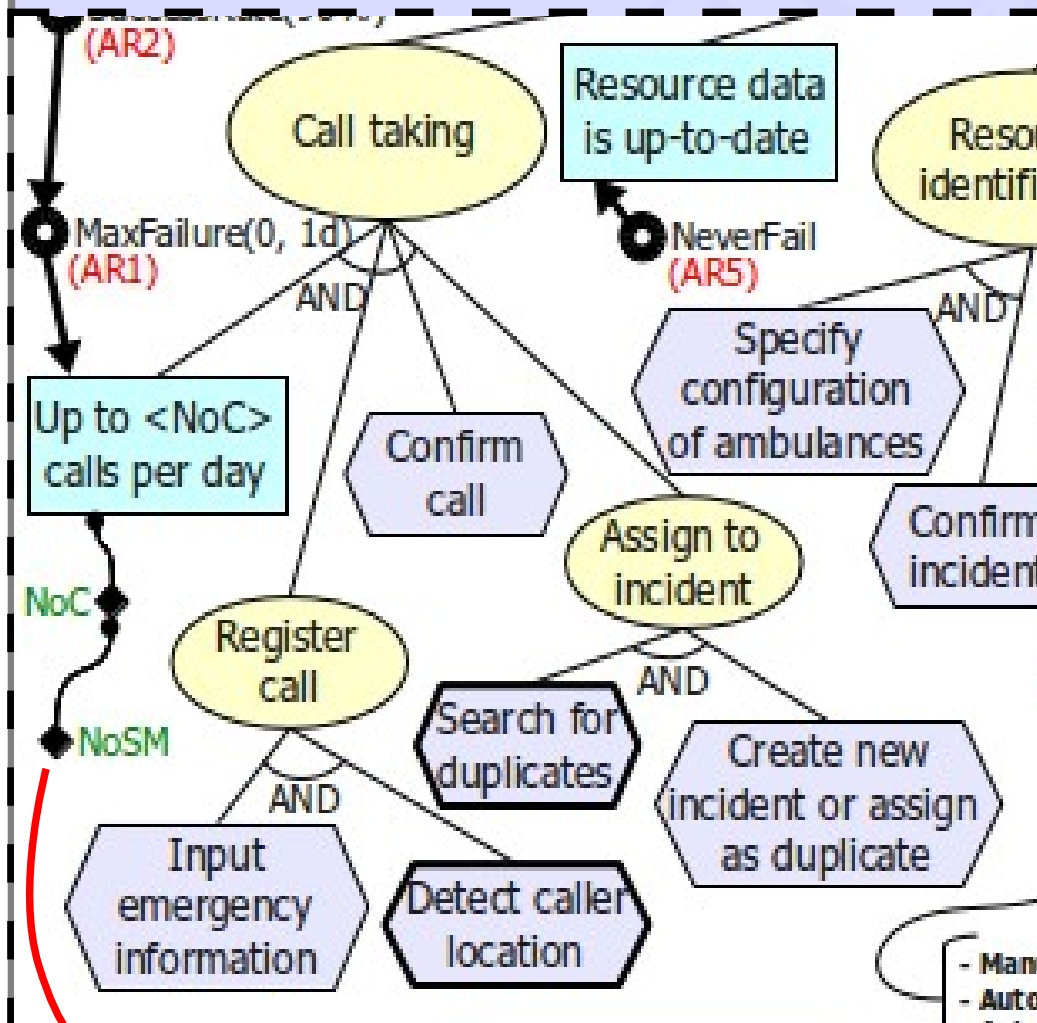
- New idea in the context of my PhD research;
 - Awareness Requirements (AwReqs)...
 - System Identification...
 - Evolution Requirements (EvolReqs)! 
- Working with John and Alexei;
- Paper to be submitted to SEAMS 2012 (deadline January 23rd).

Background: AwsReqs

- **AwReqs determine the critical requirements. We should be aware of their success/failure;**



Background: SysId



$$\Delta(\text{AR1} / \text{NoSM})[0, \text{MaxSM}] > 0$$

- System Identification:
 - Indicator/parameter qualitative relations;
 - Information to reason over during reconfig.
- Reconfiguration:
 - Search on the solution space;
 - Algorithms are work in progress...

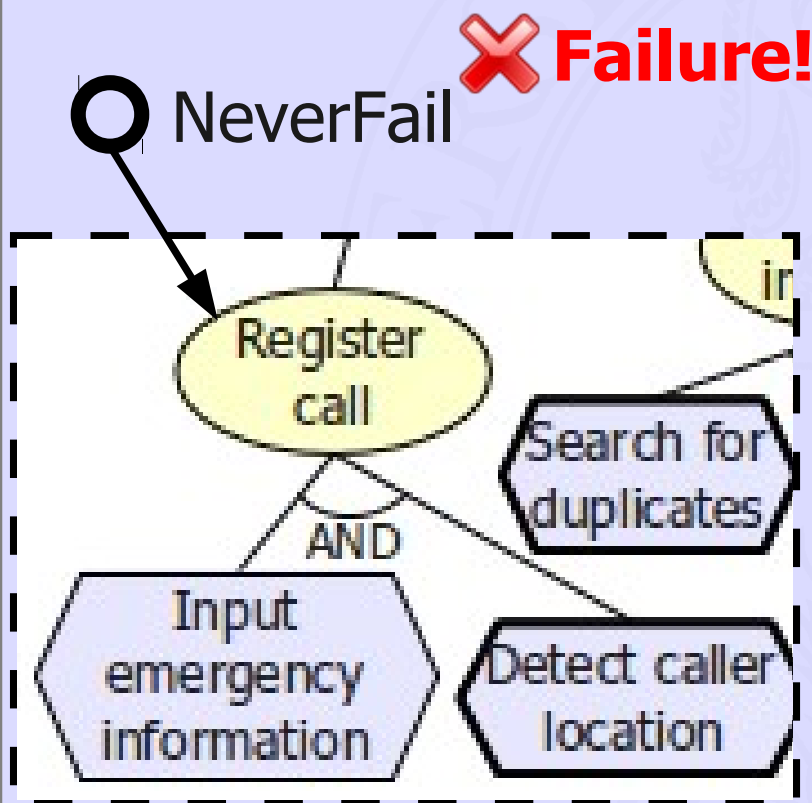
Motivation

- Is reconfiguration enough?
- Moreover, is it what stakeholders want?
- Some other strategies:
 - Try again after some period of time;
 - Warn / delegate the goal/task to an actor;
 - Relax (strengthen) a requirement;
 - Redundancy;
 - Etc...

Motivating example

- Possible strategies:

- Try again (it could be a temporary glitch in the input submission);
- Relax the goal, by disabling caller detection (we can do without it);
- Delegate the top goal to a staff member (switch to manual dispatch);
- Abort (fail gracefully).

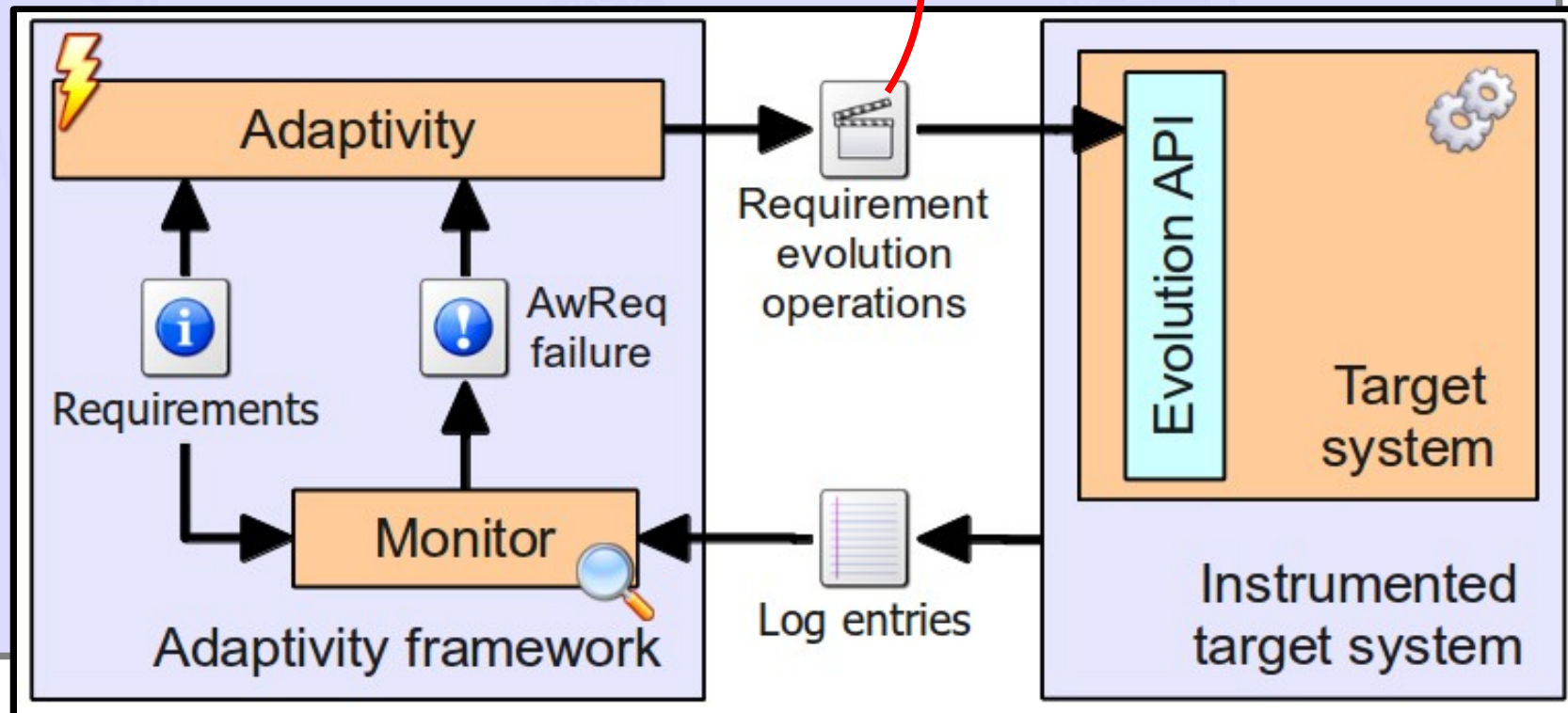


Proposal: Evolution Requirements

Retry:



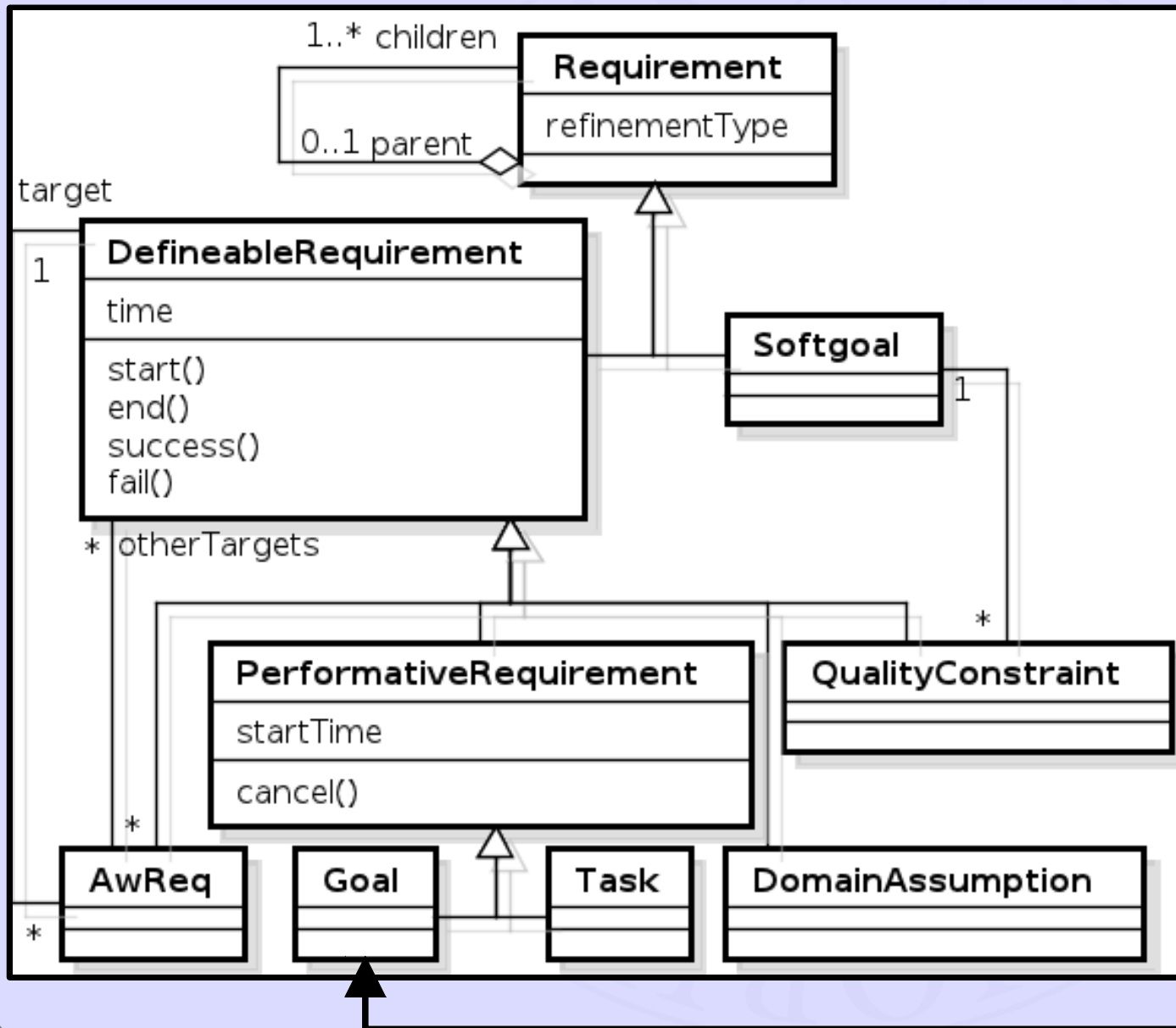
```
g' = new-instance(G_RegCall);  
copy-data(g, g');  
terminate(g);  
rollback(g);  
wait(5s);  
initiate(g');
```



Primitive operations:

- abort(ar);
- apply-config(C, L);
- change-param([R|r], p, v);
- copy-data(r, r');
- disable(R);
- enable(R);
- find-config(algo, ar);
- initiate(r);
- new-instance(R);
- resume(r);
- rollback(r);
- send-warning(A, ar);
- suspend(r);
- terminate(r);
- wait(t);
- wait-for-fix(ar).

Base model for requirements



G_RegCall

etc...

Patterns form Adaptation Strategies

```
Retry(copy : boolean = true; time : long) {  
    r = awreq.target;  
    R = r.class;  
    r' = new-instance(R);  
    if (copy) copy-data(r, r');  
    terminate(r);  
    rollback(r);  
    wait(time);  
    initiate(r');  
}
```

```
Delegate(a : Actor) {  
    send-warning(a, awreq);  
    wait-for-fix(awreq);  
}
```

More patterns...

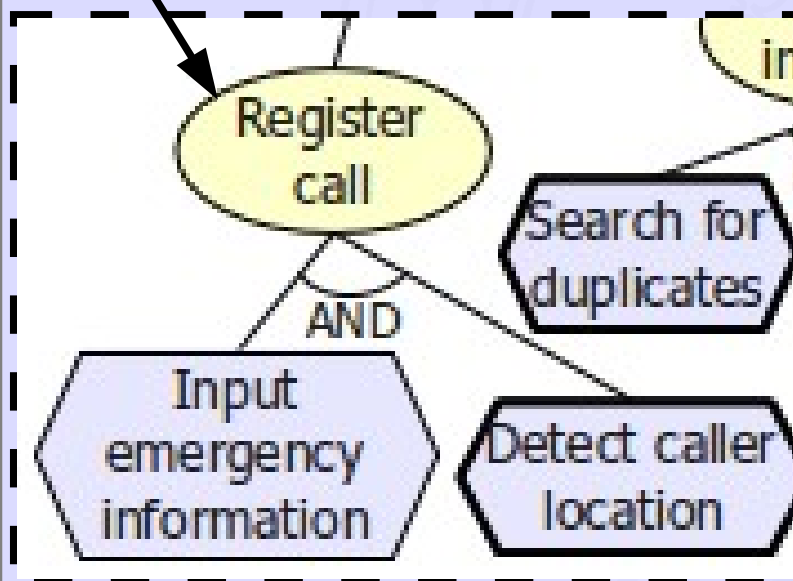
```
RelaxDisableChild(r : Requirement = awreq.target;  
level : Level = INSTANCE; childClass : Class) {  
    if (level == CLASS) {  
        disable(childClass);  
    }  
    else {  
        suspend(r);  
        for (child in r.children) {  
            if (child.class == childClass) {  
                terminate(child);  
                rollback(child);  
            }  
        }  
        resume(r);  
    }  
}
```

Patterns simplify the specification

- Associated to this AwReq:

- Retry(5000);
- RelaxDisableChild(T_DetectLoc)
- Delegate(G_GenDispatch, StaffMember)
- Abort().

○ NeverFail

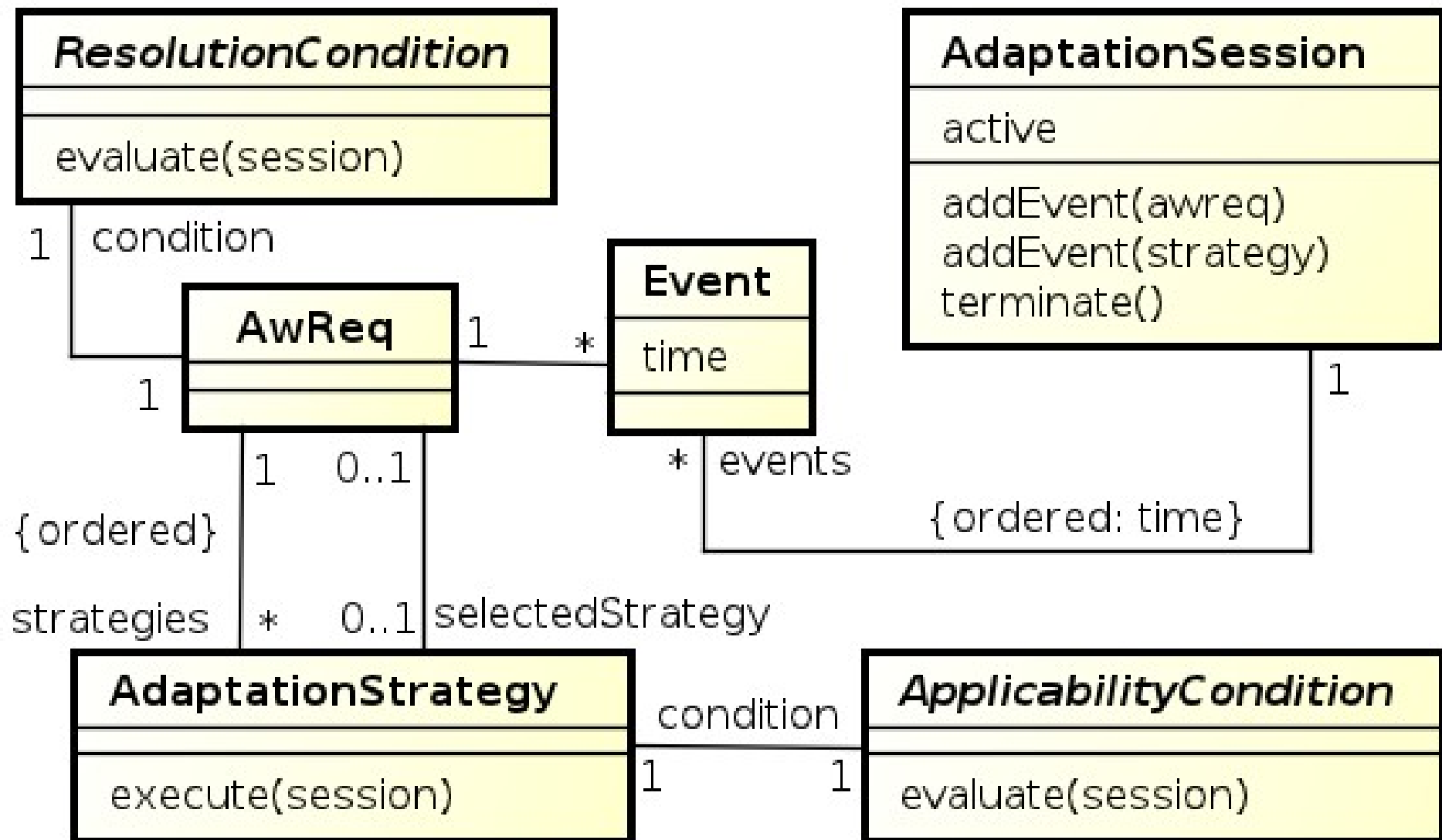


Reconfiguration as a strategy

```
Reconfigure(algo : FindConfigAlgorithm,  
ar : AwReq, level : Level) {  
  
    C' = find-config(algo, ar)  
    apply-config(C', level)  
  
}
```

Algorithms are subject of future
work... (target: RE 2012)

ECA-based coordination process (1)



ECA-based coordination process (2)

```
processEvent(ar : AwReq) {  
    session = findOrCreateSession(ar.class);  
    session.addEvent(ar);  
    solved = ar.condition.evaluate(session);  
    if (solved) break;  
  
    ar.selectedStrategy = null;  
    for each s in ar.strategies {  
        appl = s.condition.evaluate(session);  
        if (appl) {  
            ar.selectedStrategy = s;  
            break;  
        }  
    }  
  
    if (ar.selectedStrategy == null)  
        ar.selectedStrategy = ABORT;  
  
    ar.selectedStrategy.execute(session);  
    ar.condition.evaluate(session);  
}
```

Different coordination processes?

- Any process that (using the model):
 - Selects the best strategy to apply;
 - Execute it;
 - Check if the problem has been solved;
 - If not, loop back.
- Might be worth investigating:
 - A preferences-based process;
 - Attach AwReqs to EvolReqs to create hierarchies of controllers;
 - Any other idea?

Thank You! Questions? Feedback?



Acknowledgment:

The research reported in this presentation was partially funded by the ERC advanced grant 267856 "Lucretius: Foundations for Software Evolution", unfolding during the period of April 2011 - March 2016.



Università degli Studi di Trento
Facoltà di Scienze Matematiche, Fisiche e Naturali
Dipartimento di Ingegneria e Scienza
dell'Informazione

(Requirement) Evolution Requirements for Adaptive Systems

Vítor E. Silva Souza

vitorsouza@disi.unitn.it
<http://disi.unitn.it/~vitorsouza>