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

Monitoring and Diagnosing Malicious Attacks with Autonomic Software

Vítor E. Silva Souza
http://www.disi.unitn.it/~vitorsouza/

John Mylopoulos
http://www.cs.toronto.edu/~jm/

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.

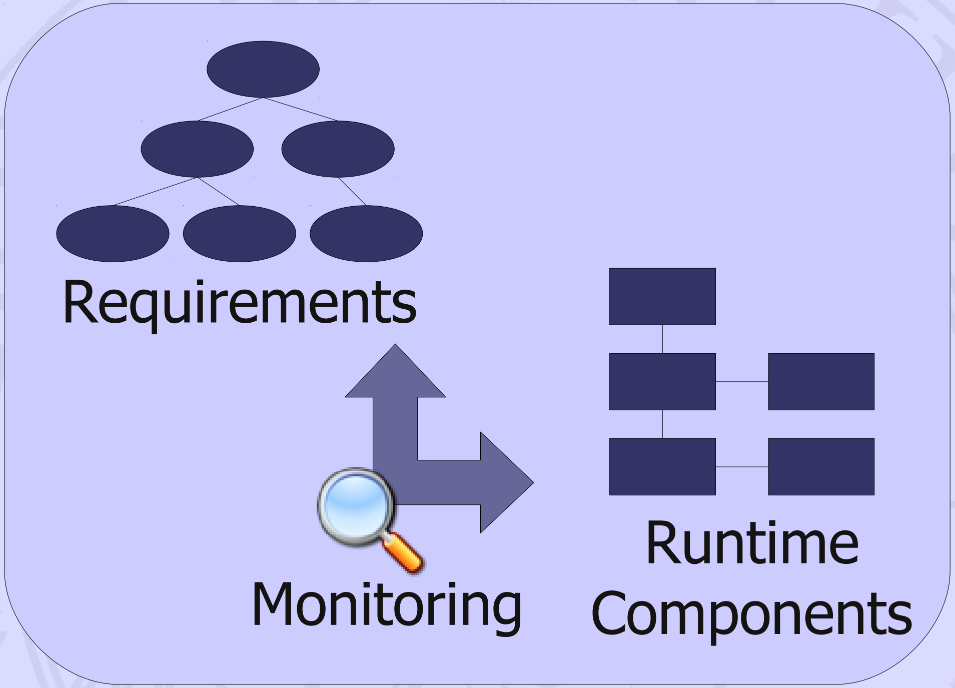
Agenda

- Motivation;
- A diagnosing framework;
- Proposed extensions for this framework:
  - Anti-goals;
  - Contextual variability;
- Evaluation;
- Conclusions.

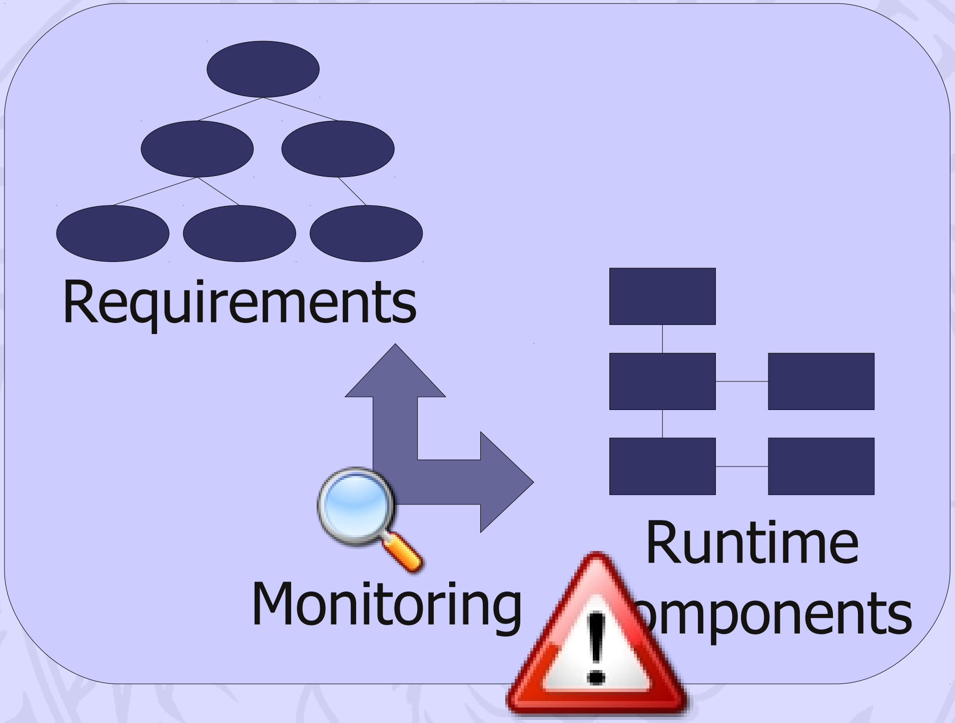
Motivation

- Monitoring and diagnosing against requirements:
  - Old problem (e.g. Fickas & Feather, 1995);
  - Considerable recent attention in the context of adaptive and autonomic software systems;
- Autonomic systems:
  - Operate on their own according to a set of rules;
  - Self-configuration, self-optimization, self-healing and self-protection;
  - Monitor (failures, sub-optimal behaviors, attacks, etc.) → diagnose → compensate.

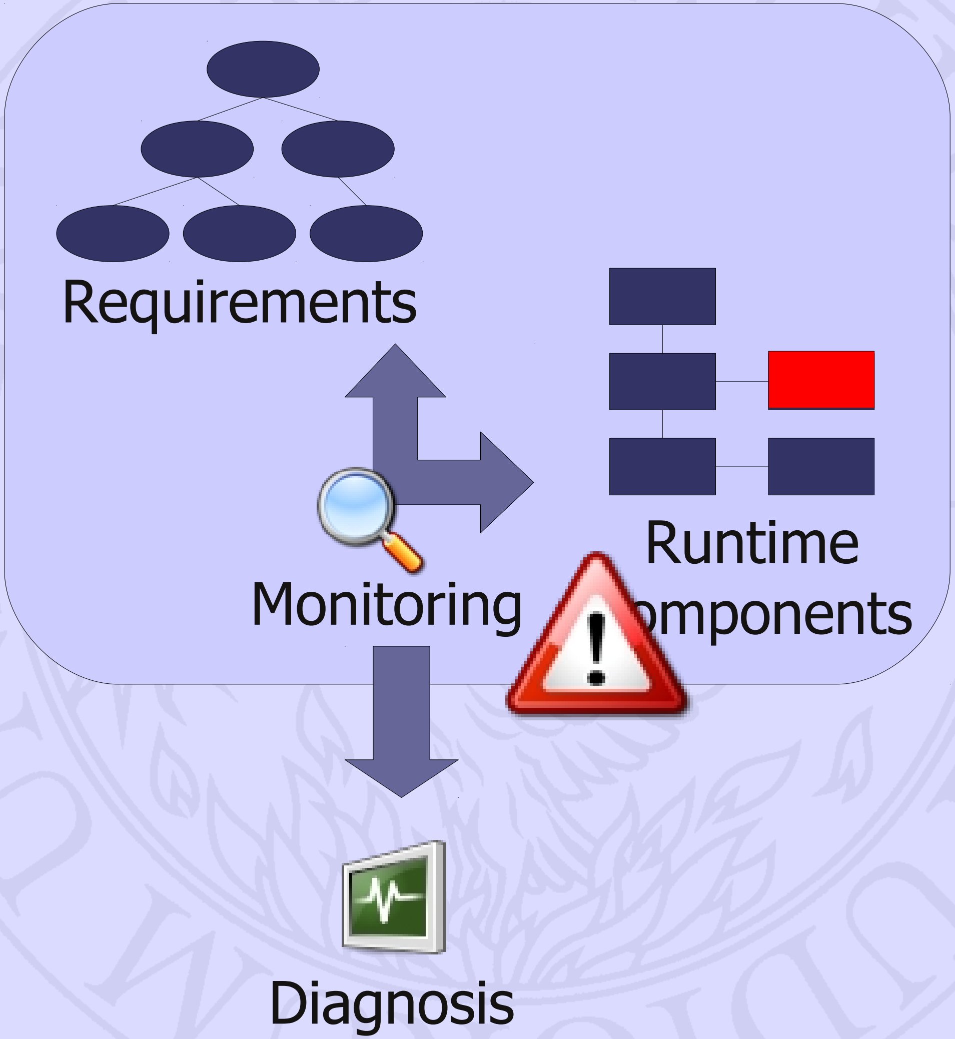
Motivating Scenario (1)



Motivating Scenario (1)



Motivating Scenario (1)



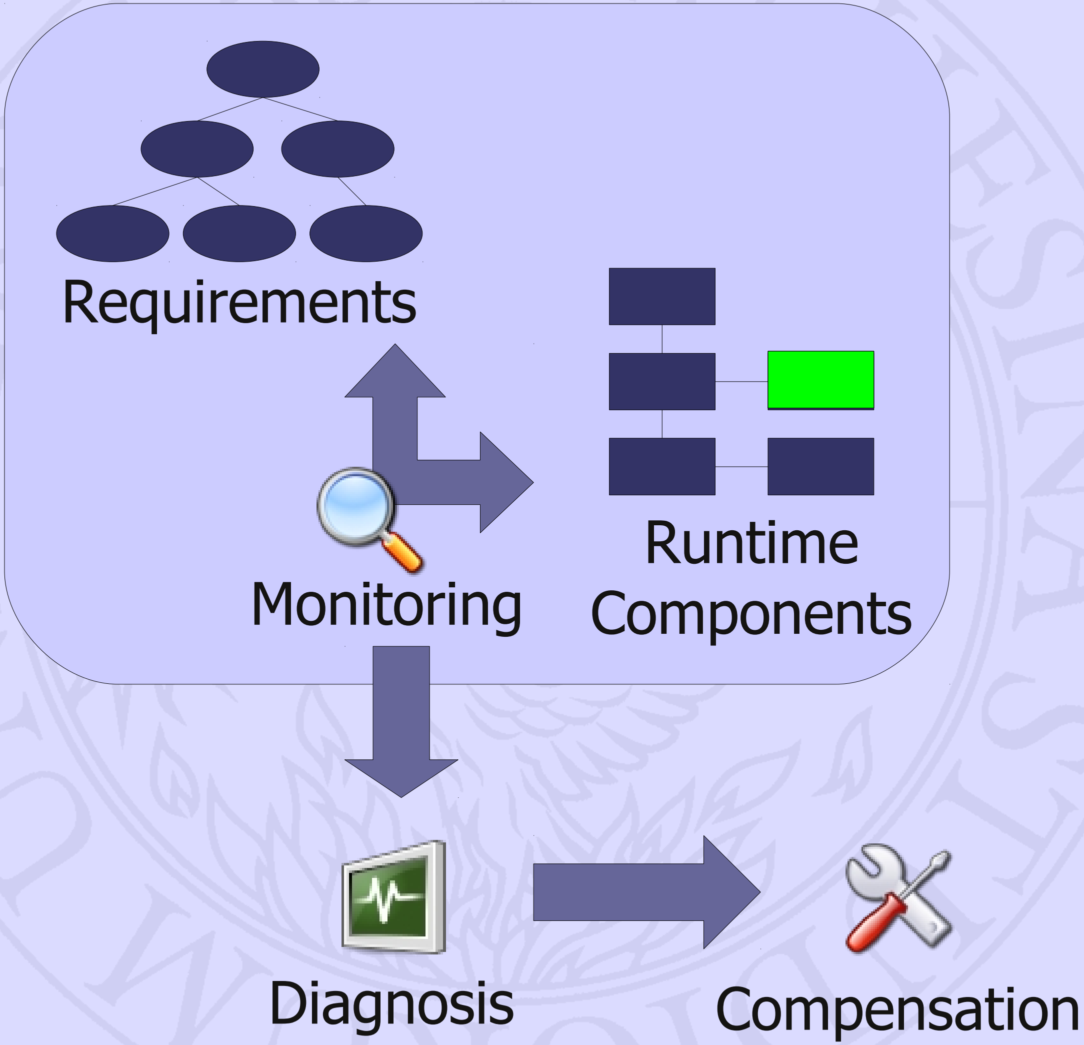
Requirements

Runtime
omponents

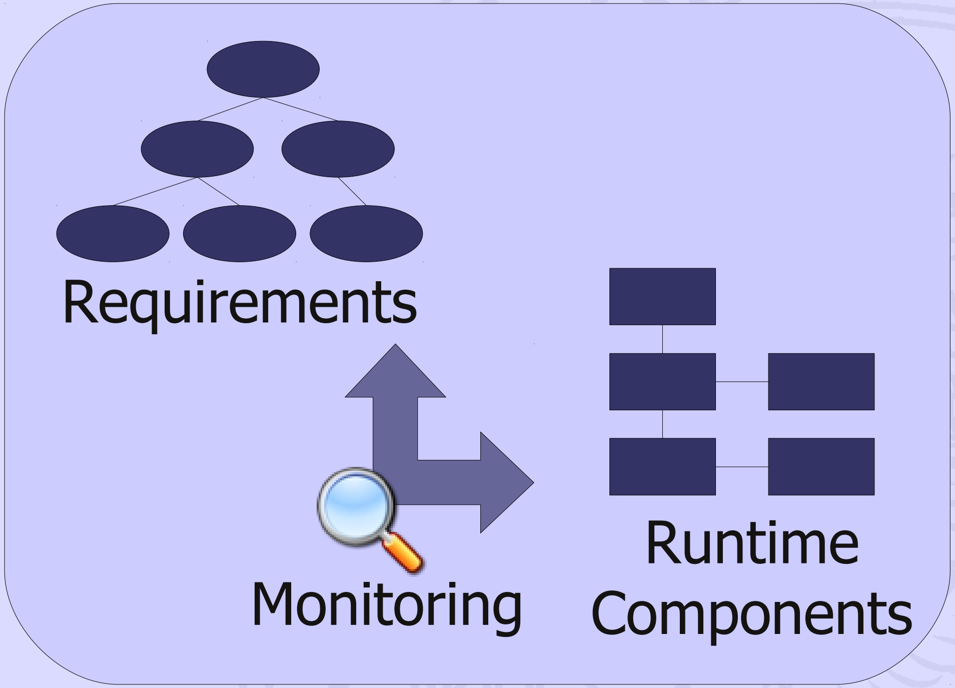
Monitoring

Diagnosis

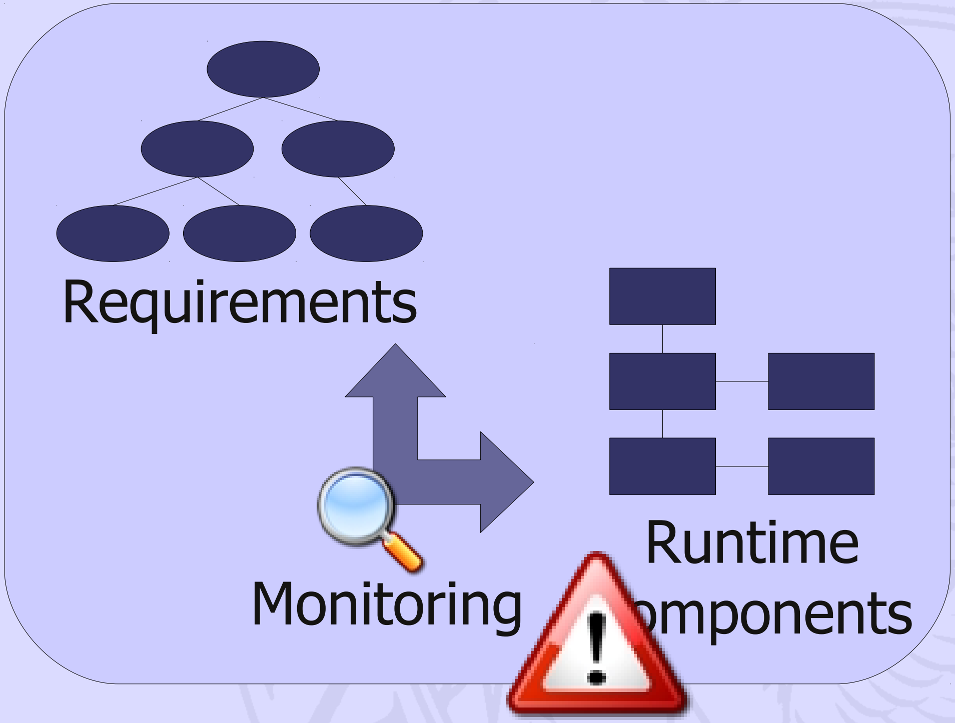
Motivating Scenario (1)



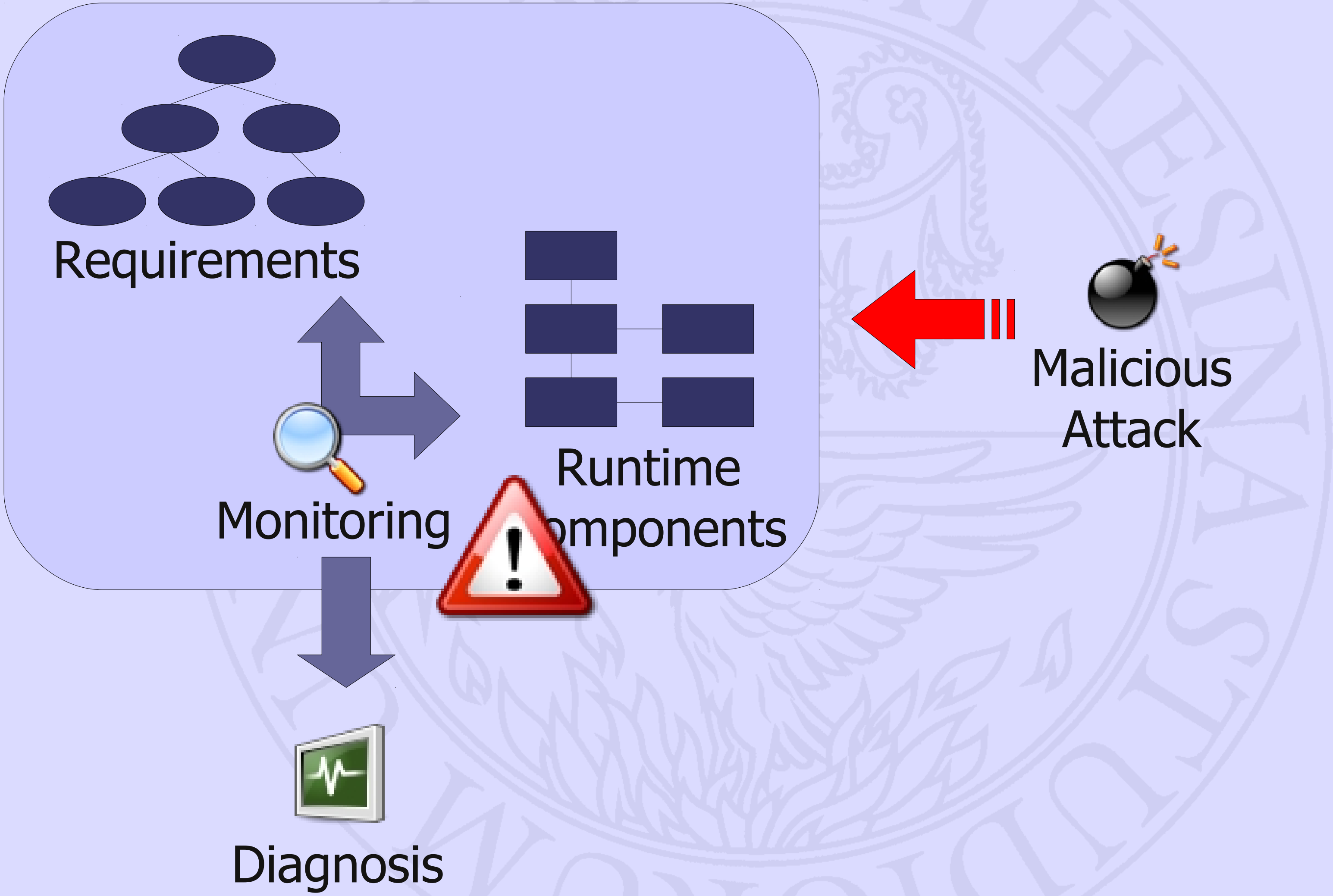
Motivating Scenario (2)



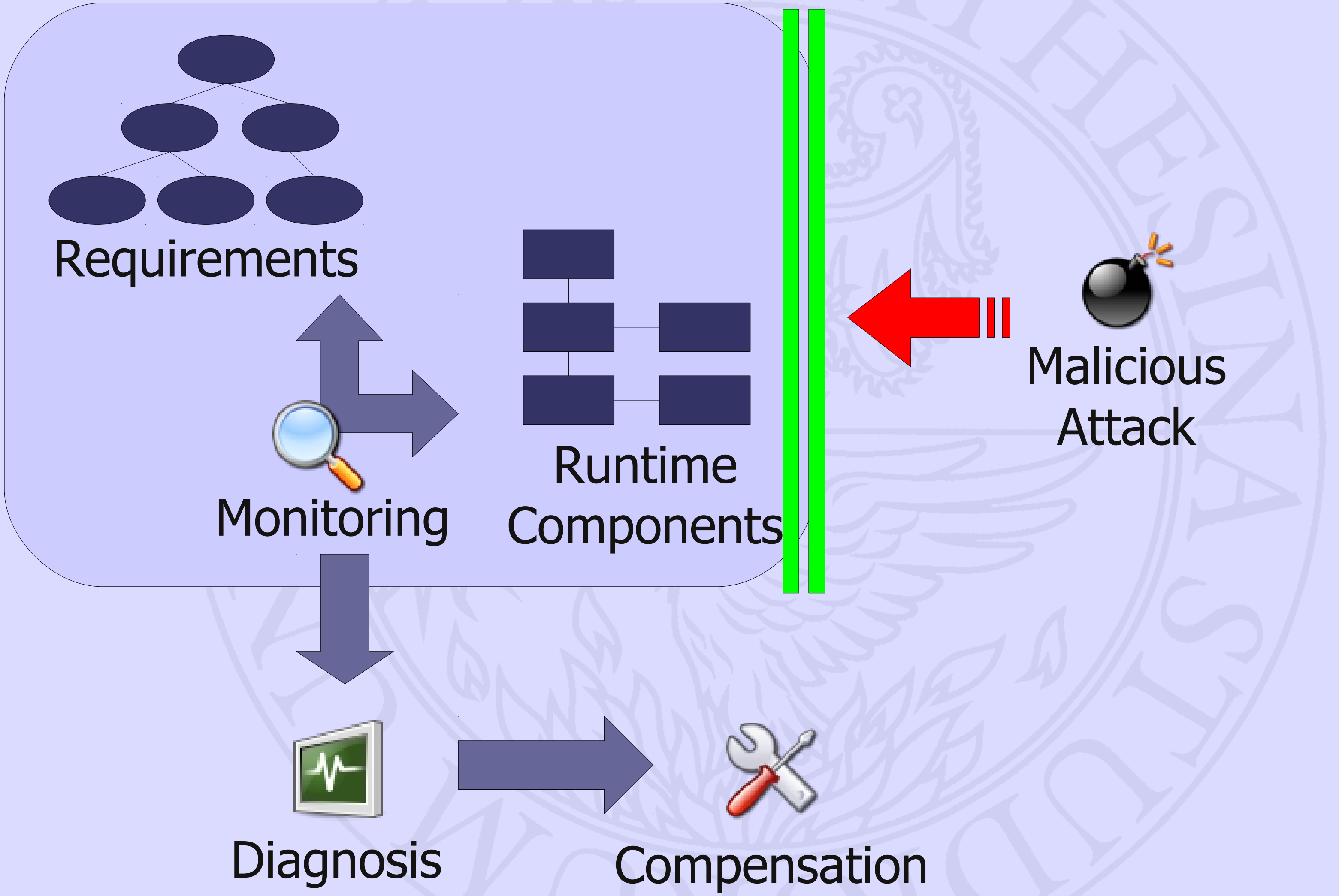
Motivating Scenario (2)



Motivating Scenario (2)



Motivating Scenario (2)



Requirements

Monitoring

Runtime Components

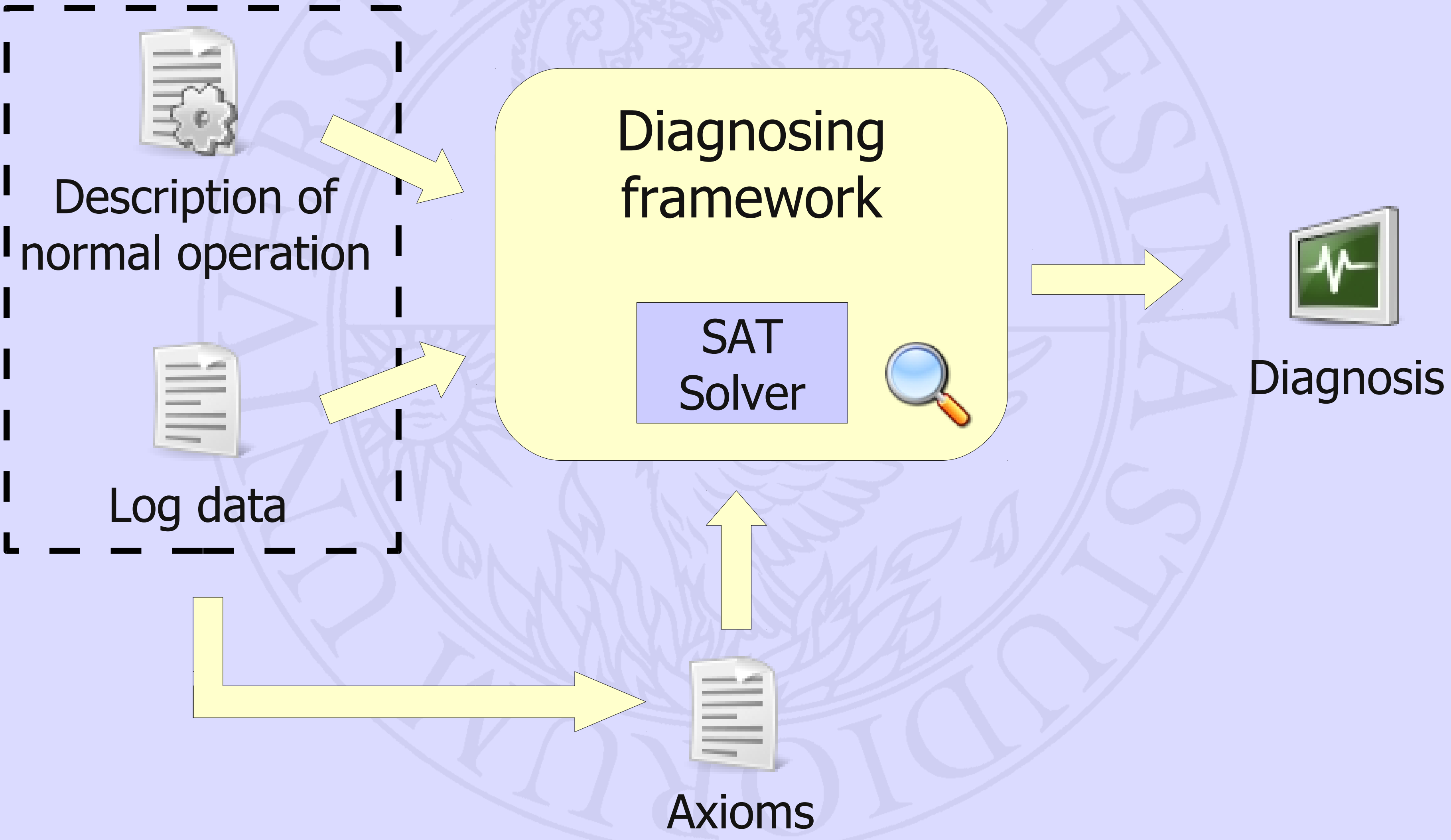
Malicious Attack

Diagnosis

Compensation

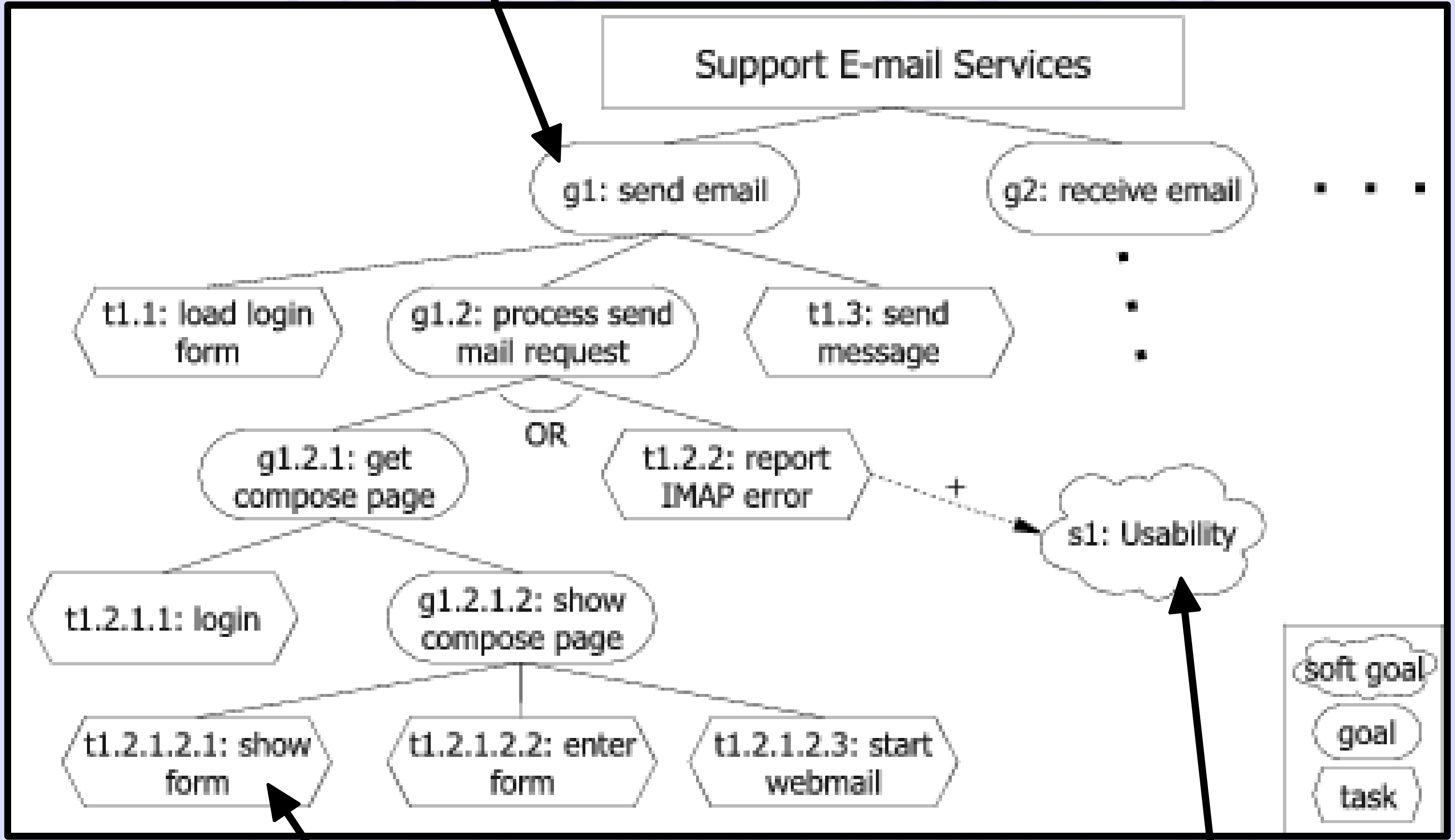
A diagnosing framework

- Based on Wang et. al.



Requirements are goal models

Main goals of the application



A monitorable piece of the software

Non-functional requirements

Diagnosing

- Based on requirements and an execution log:

**url_entered(1), occ(t1.1, 2), correct_form(3),
~ wrong_imap(4), occ(t1.2.1.1, 5), correct_key(6),
occ(t1.2.1.2.1, 7), occ(t1.2.1.2.2, 8), occ(t1.2.1.2.3, 9),
~ webmail_started(10), occ(t1.3, 11), ~ email_sent(12)**

- The framework produces:
  - Facts;
  - Propagation axioms;
  - Contribution axioms;
  - Deniability axioms.

fd(t1.2.1.1); fd(t1.3)
fd(t1.2.1.2); fd(t1.3)
fd(t1.2.1.3); fd(t1.3)

- The SAT solver then derives the diagnosis.

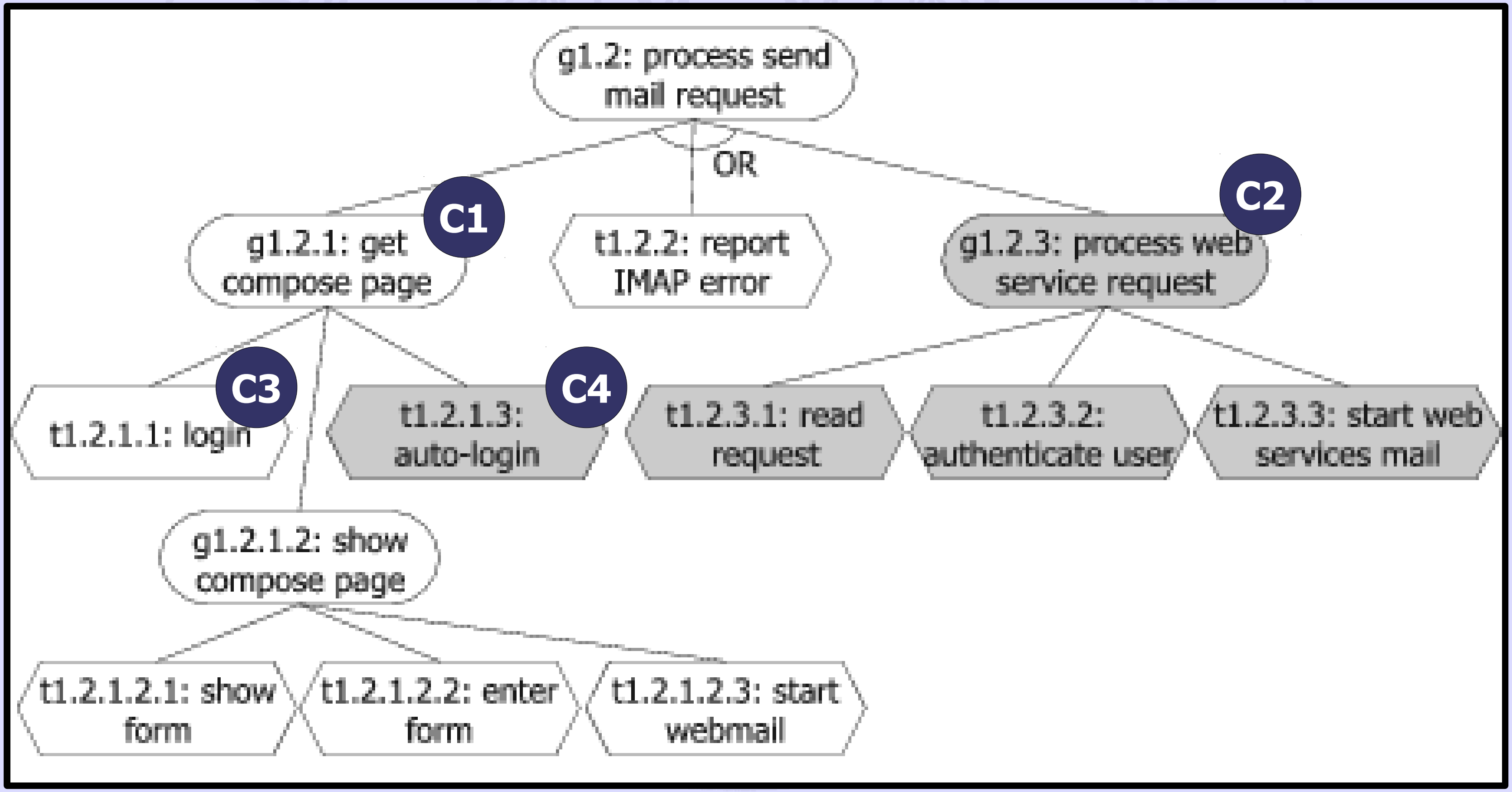
Proposed extensions for the framework

- Goal: to monitor and diagnose malicious attacks;
- Add support for anti-goals[1]:
  - Software components are not faulty;
  - Problem caused by an external agent;
- Add support for contextual variability[2]:
  - Attacks are notoriously context-dependent;
  - Richer goal model.

[1] – Based on anti-goals proposal by Lamsweerde et al.;
[2] – Based on contextual variability proposed by Lapouchnian.

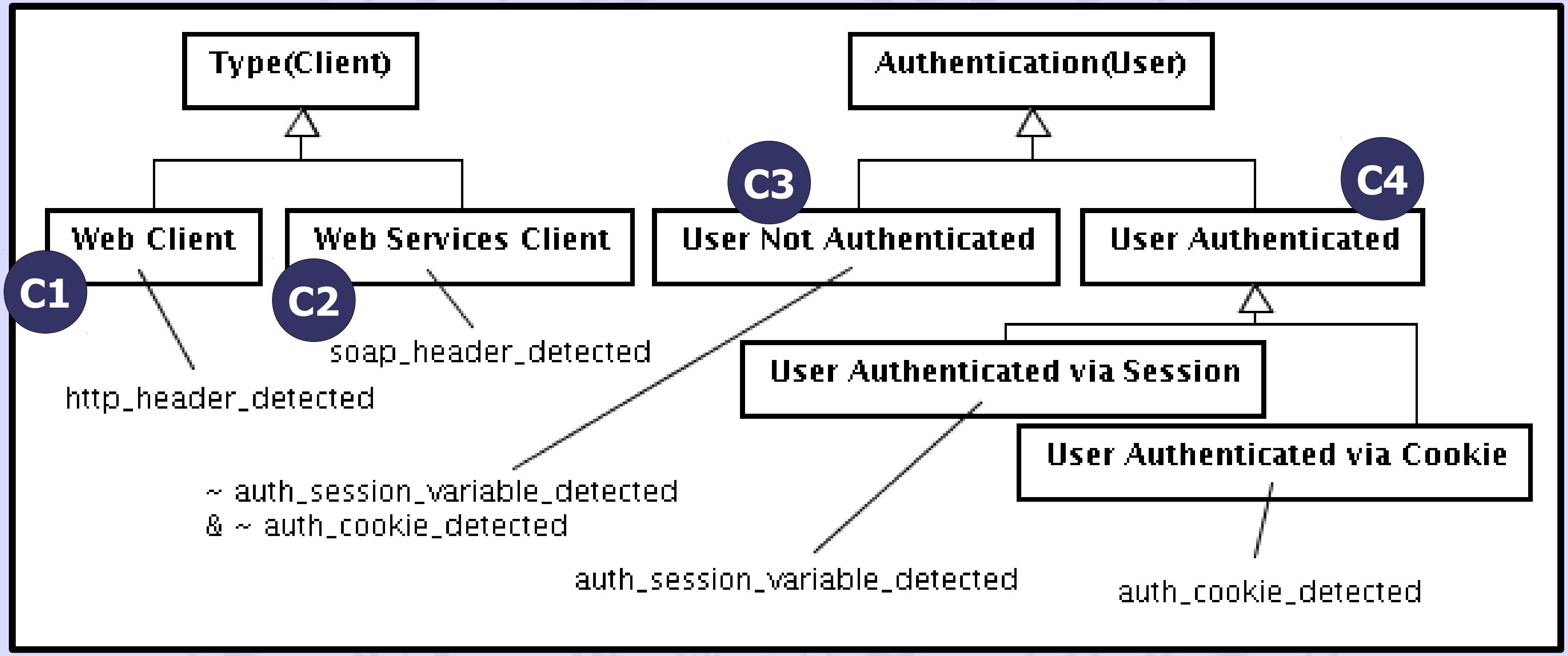
Support for contextual variability

- Goals and tasks can be annotated with context;
- Instrumented code logs data needed to identify active context.


g1.2: process send mail request
OR
C1
g1.2.1: get compose page
t1.2.2: report IMAP error
C2
g1.2.3: process web service request
C3
t1.2.1.1: login
C4
t1.2.1.3: auto-login
t1.2.3.1: read request
t1.2.3.2: authenticate user
t1.2.3.3: start web services mail
g1.2.1.2: show compose page
t1.2.1.2.1: show form
t1.2.1.2.2: enter form
t1.2.1.2.3: start webmail

Support for contextual variability

- Contexts can form hierarchies:



- A context is active if:
  - A sub-context is active;
  - Its formula evaluates to true.

Support for contextual variability

- Contextual information affect:
  - Satisfiability propagation (an AND goal is satisfied iff all its children **with active context** are satisfied);
  - Task/goal occurrence – tasks and goals cannot occur when their context is inactive:
    - Instrumented code not logging context information;
    - Software not according to specifications.

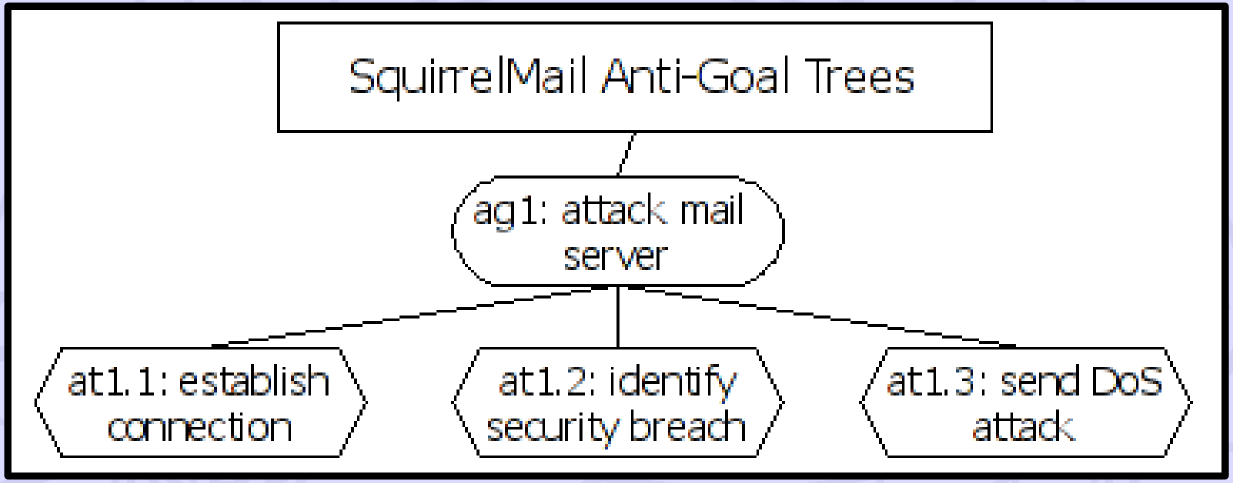
$$occ(g,t_s,t_e) \land \neg context\_formula(g,t_s) \rightarrow iocc(g,s)$$

$$occ(a,t_{occ}) \land \neg context\_formula(a,t_{occ}) \rightarrow iocc(a,s)$$

Context formula is built navigating context hierarchy depth-first and joining the leaf-contexts in a disjunction.

Support for anti-goals

- Goal models can include anti-goal trees;

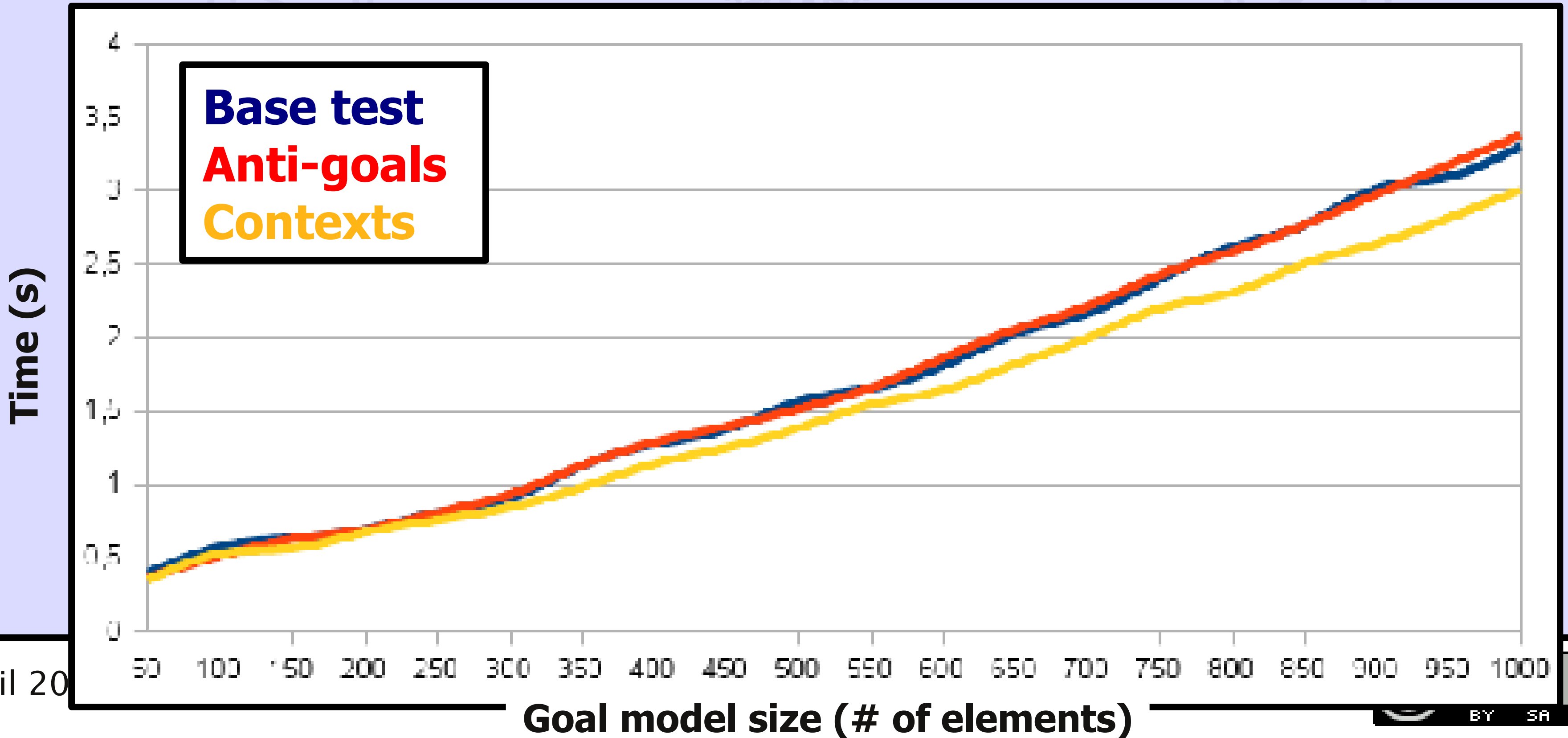

SquirrelMail Anti-Goal Trees
ag1: attack mail server
at1.1: establish connection
at1.2: identify security breach
at1.3: send DoS attack

- Anti-goals and tasks have sets of targets;
- Attack detection software produces log information on anti-goals;
- Anti-goal satisfiability axioms are produced:

$$\forall e \in \{e_1, e_2, \ldots, e_n\} : occ(a, t_s, t_e) \wedge fd(e, s) \rightarrow fs(a, s)$$

Performance evaluation

- Used the ATM Simulation by R. C. Bjork;
  - Goal model and log files are replicated in different sizes (from 100 to 1000);
  - New features maintain framework scalability:



Conclusions

- Contributions:
  - Novel approach for M&D malicious attacks;
  - Support for goal models enriched with contexts;
  - Preliminary feasibility and scalability tests.
- Future work:
  - Study of possible compensating mechanisms;
  - Complementing the diagnostic reasoner with probabilistic techniques;
  - Further research on autonomic computing, with emphasis on self-protection.

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

Monitoring and Diagnosing Malicious Attacks with Autonomic Software

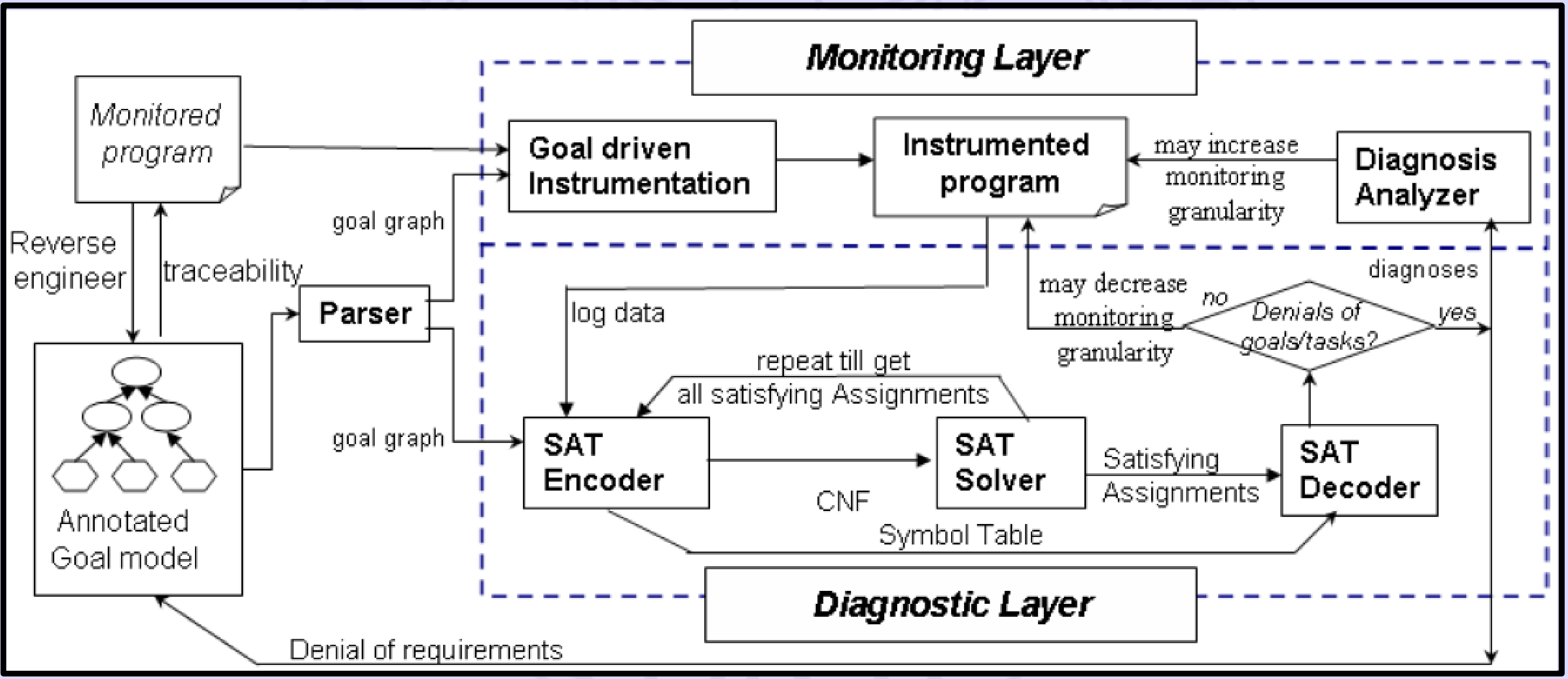
Vítor E. Silva Souza
http://www.disi.unitn.it/~vitorsouza/

John Mylopoulos
http://www.cs.toronto.edu/~jm/

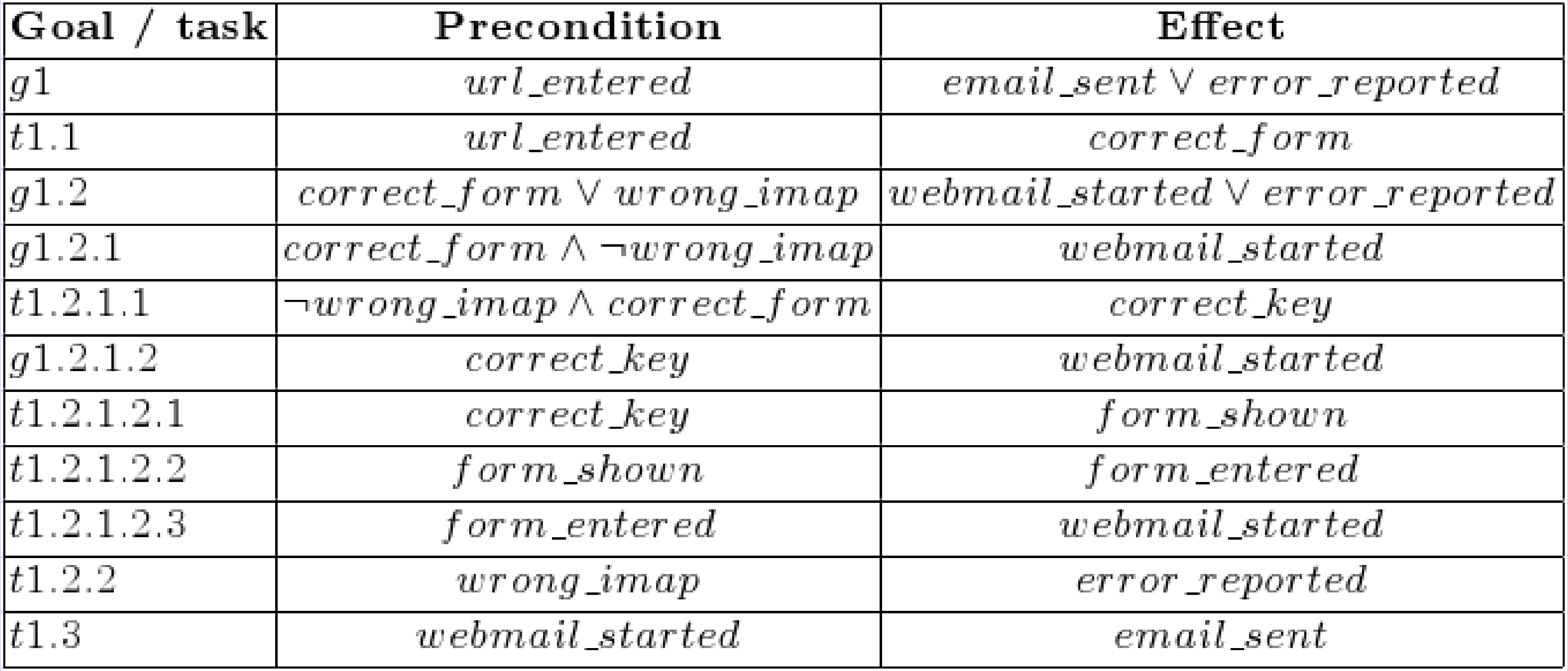
Photo is Copyright © 2008 Andrea Caranti (http://www.flickr.com/photos/11102419@N00/1476590956/). Used with author explicit permission.

Diagnosing framework by Wang et al.

A general monitoring framework, paired with a SAT-based diagnostic reasoner adapted from Artificial Intelligence (AI) theories of action and diagnosis.



Monitoring the specification



Goal / task	Precondition	Effect
$g1$	$url\_entered$	$email\_sent \lor error\_reported$
$t1.1$	$url\_entered$	$correct\_form$
$g1.2$	$correct\_form \lor wrong\_imap$	$webmail\_started \lor error\_reported$
$g1.2.1$	$correct\_form \land \neg wrong\_imap$	$webmail\_started$
$t1.2.1.1$	$\neg wrong\_imap \land correct\_form$	$correct\_key$
$g1.2.1.2$	$correct\_key$	$webmail\_started$
$t1.2.1.2.1$	$correct\_key$	$form\_shown$
$t1.2.1.2.2$	$form\_shown$	$form\_entered$
$t1.2.1.2.3$	$form\_entered$	$webmail\_started$
$t1.2.2$	$wrong\_imap$	$error\_reported$
$t1.3$	$webmail\_started$	$email\_sent$

Can turn monitoring on or off

Must be true before a task is executed or a goal is satisfied

Must be true after a task is executed or a goal is satisfied