The Evolution of Tropos: Contexts, Commitments and Adaptivity

Raian Ali, Amit K. Chopra, Fabiano Dalpiaz, Paolo Giorgini, John Mylopoulos, and Vitor E. Silva Souza

June 7th, 2010

UNIVERSITY
OF TRENTO - Italy
Information Engineering
and Computer Science Department

i* '10
FOURTH
INTERNATIONAL
I* WORKSHOP
istar 2010 - 07-08 JUNE 2010
HAMMAMET, TUNISIA

This talk's tag cloud

requirements
commitments
goal
systems
goals
actors
models
reasoning
context
contextual
agent's
agents
awreq
system
social
model
research
modeling
support
security
changes
agent
Tropos
runtime
settings
example
evolution
internals
actor
software
variants
external
architecture
awareness
report
Social
iii
ii
re
respect
framework
activities
human
techniques
self-reconfigurable
within
UniTN
response
protocol
refer
plan
something
AwReq
capture
interested
autonomy
among
loop
ing
main
meet
developing
behavior
dependencies
computing
istar
important
feedback
sociotechnical
internal
STS
others
new
actions
achieve
meeting
able
participants
environment
contexts

Created via http://www.wordle.net/
.

Software evolution (*à la* UniTn)

- We aim at a comprehensive approach to software evolution based on requirements
- The systems we have in mind have to
  - be aware of their own requirements
  - consider the influence of the surrounding context
  - take into account social relations with other systems
  - be adaptive

What's in our approach?

- Four basic techniques
  1) Contextual requirements
  2) Modeling applications with social commitments
  3) Adaptive sociotechnical systems
  4) Requirements awareness

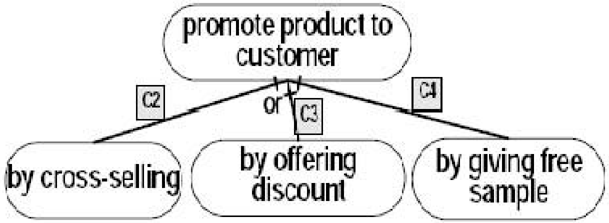
Contextual requirements

Contact: Raian Ali (ali@disi.unitn.it)

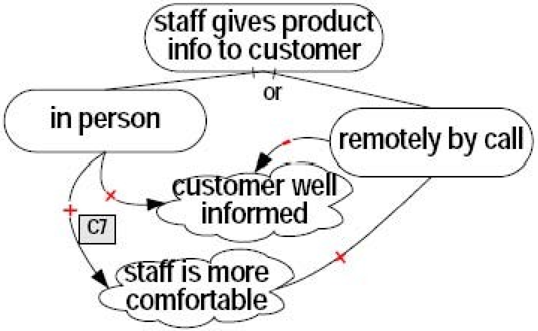
Contextual Goal Model

- Goals answer Why in requirements, not When/Where
  - Context to the rescue
- Context influences humans before software:
  - Software has to reflect human adaptation to context
- Example: if a tourist hasn't had lunch yet and it's lunch time, a tour guide has to find a restaurant
  - If the tourist is vegetarian, some alternatives will be ruled out

Contextual Variation Points



OR-Decomposition

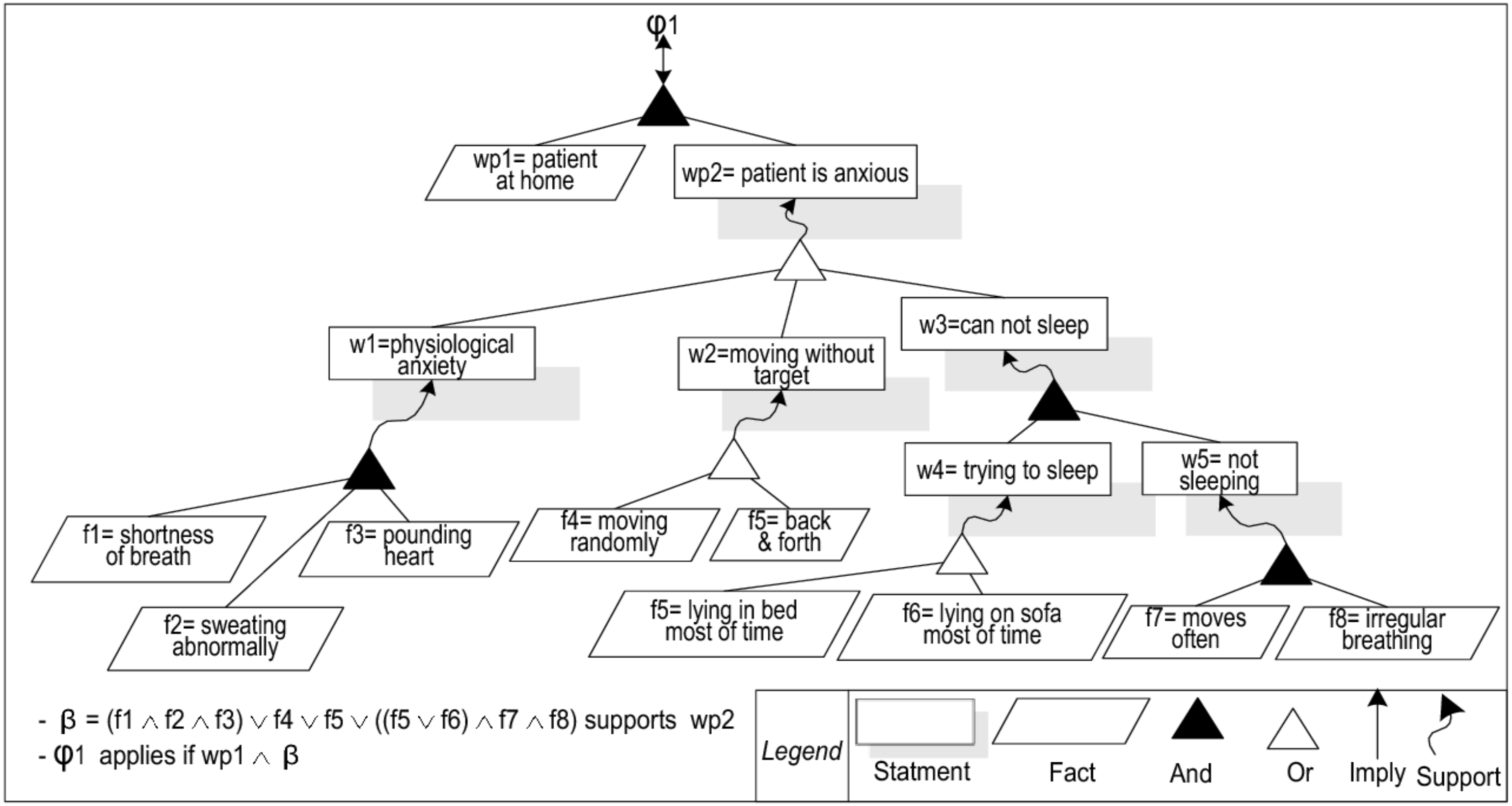

staff gives product info to customer
or
in person
remotely by call
customer well informed
C7
staff is more comfortable

Contribution

Dependency
Root goals
AND-Decomposition
Means-End

Context Analysis

- While goal is a state of the world to reach, context is a state of the world that is the case
- Context analysis serves to know what to verify to judge if a certain context holds



- $\beta = (f1 \wedge f2 \wedge f3) \vee f4 \vee f5 \vee ((f5 \vee f6) \wedge f7 \wedge f8)$ supports wp2
- $\varphi 1$ applies if $wp1 \wedge \beta$

Automated Analysis

Analysis provided by our prototype CASE tool:

- Consistency
  - Context specification
  - Resource usage conflicts
- Derivation of variants
  - For one specific context vs. for all contexts
  - With minimum development cost

Modeling applications with social commitments

Contact: Amit K. Chopra (chopra@disi.unitn.it)

“I don’t know anything about Mr. Fitzpatrick,” repeated Mrs. Kearney. “I have my contract, and I intend to see that it is carried out.”

James Joyce, A Mother, Dubliners

The *i** framework

- Dependencies emphasizes social nature of requirements
  - Agents depend on one another
- Formally, depends(A,B,g)
  - A wants g
  - B is committed and able to deliver g

Doesn't work well for open systems (e.g. eBay)

Why? Not as social as we need!

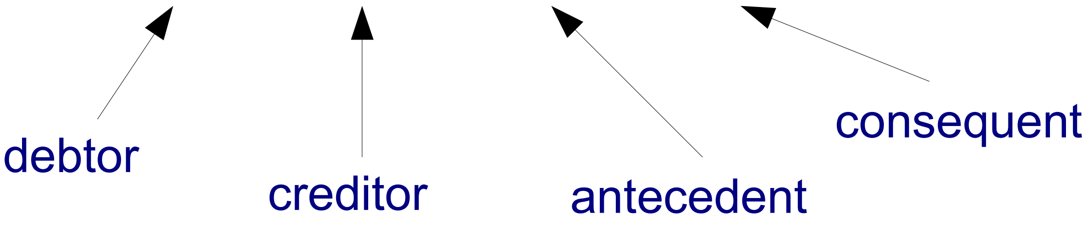
Limitations of *i** dependencies

- Refer to agent internals (recall able to)
  - A bidder does not know whether a seller has the ability to deliver
- The workability of a dependency must be justified
  - Commitment to present a paper is taken at face value
- Gives no account for interaction
  - How are dependencies established?

*i** does not cleanly separate the social
(public) from the intentional (private)

Social commitments (1)

- Agent communication is meaningful
  - Meaning in terms of commitments
  - Meaning often specified for a particular context
- For example, an offer means
  - C(seller, buyer, paid, delivered)



debtor

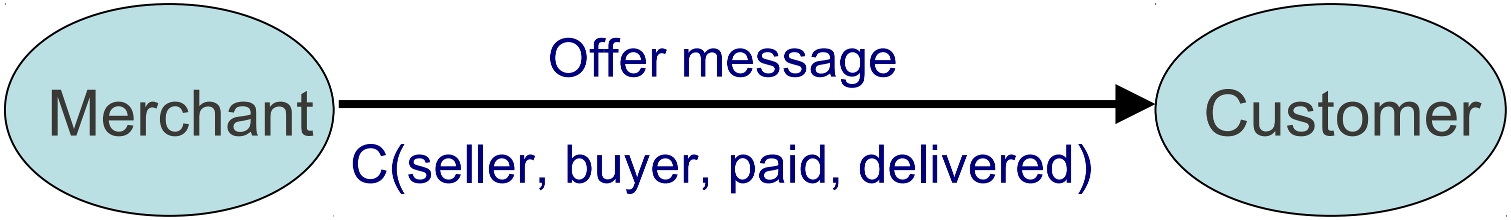
creditor

antecedent

consequent

Social commitments (2)

A social commitment does not imply any goal, intention on part of the agents



goal(merchant,delivered)?
intention(merchant,delivered)?
able(merchant,delivered)?
Maybe goal(merchant,takePaymentAndRun)!

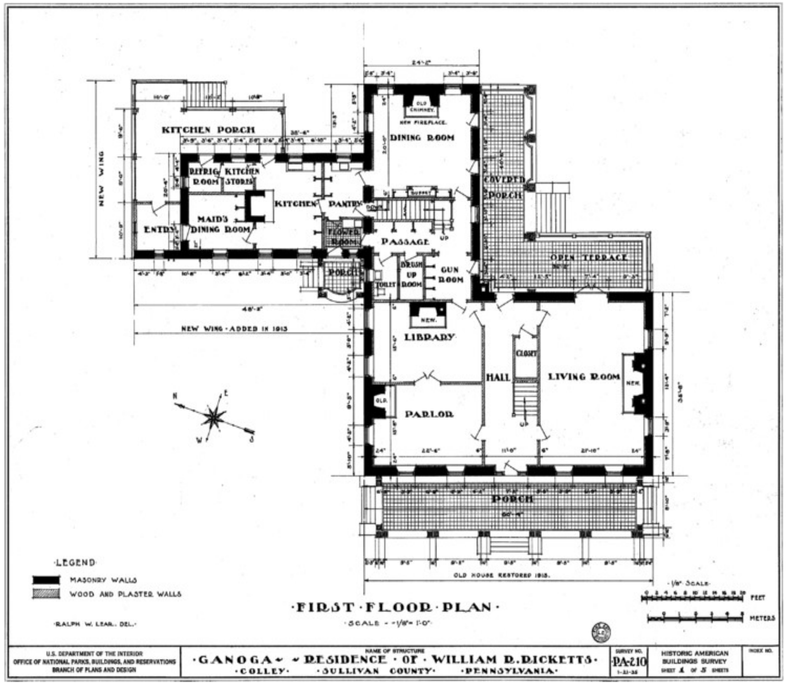
How to use social commitments?

Enable modular reasoning

- First, an agent may reason about the communications at the level of roles
  - Talk on Wednesday in Session 3 at 2:30PM
- Then, an agent may use judgments about which specific agents to interact with based on its beliefs about them

Adaptive sociotechnical systems

Contact: Fabiano Dalpiaz (dalpiaz@disi.unitn.it)



Runtime Adaptation: why?

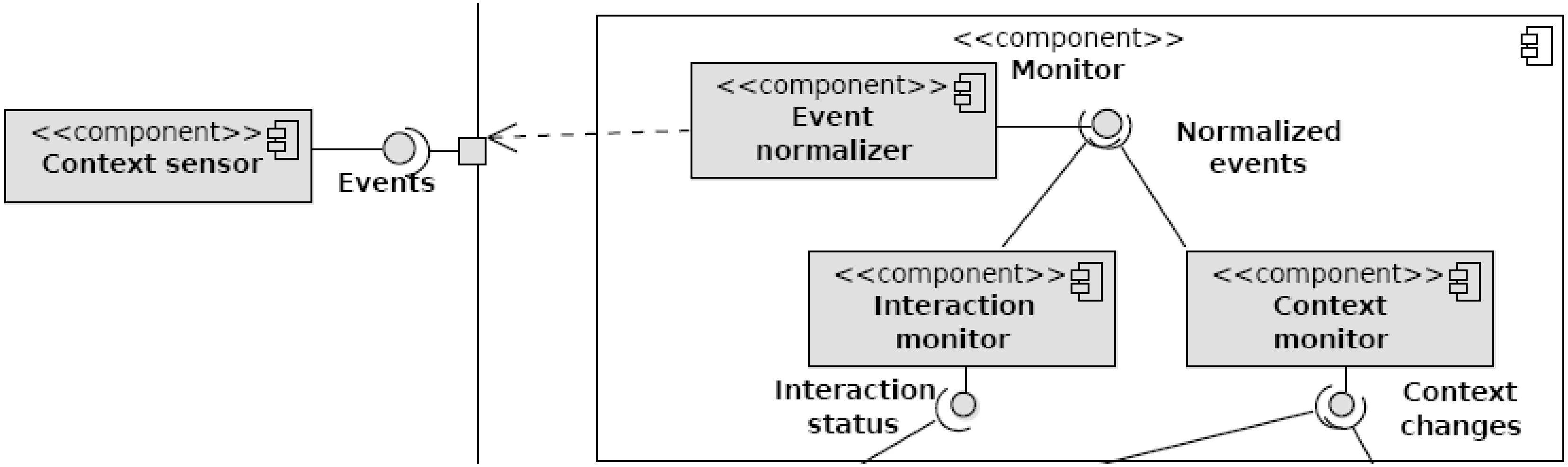
- Approaches to design adaptive software are not sufficient
- At runtime
  - Unexpected events happen
  - The system might not work as designed (bugs)
  - Some business partner might prove to be unreliable
- The solution is in the system architecture!

Our model-based architecture

- for sociotechnical systems (STS)
  - interacting socio and technical agents
- desired agent behaviour via requirements models
  - Extended goal models (context, parametric goals, activation and fulfillment conditions, timeouts)
  - Domain assumptions
- based on a monitor-diagnose-reconcile-compensate cycle
- compensation takes into account agents autonomy

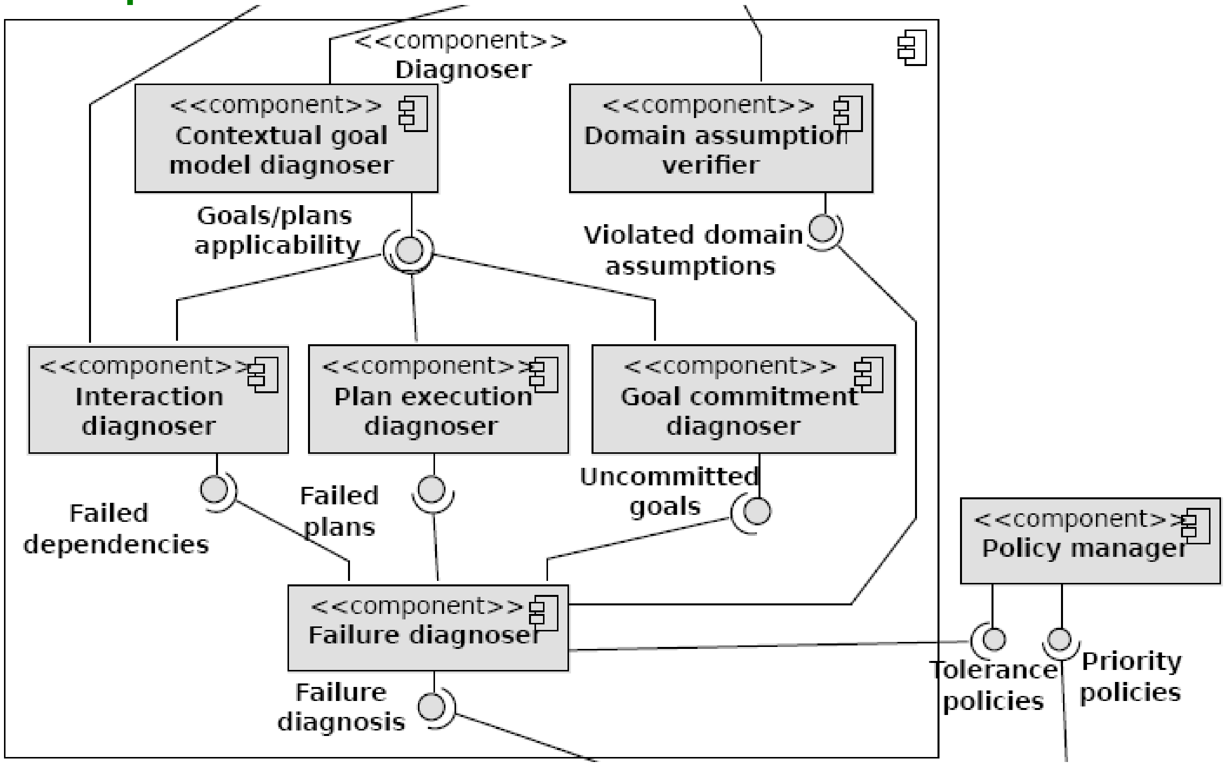
Monitoring component

- The architecture monitors interaction and changes in the context



Diagnosis component

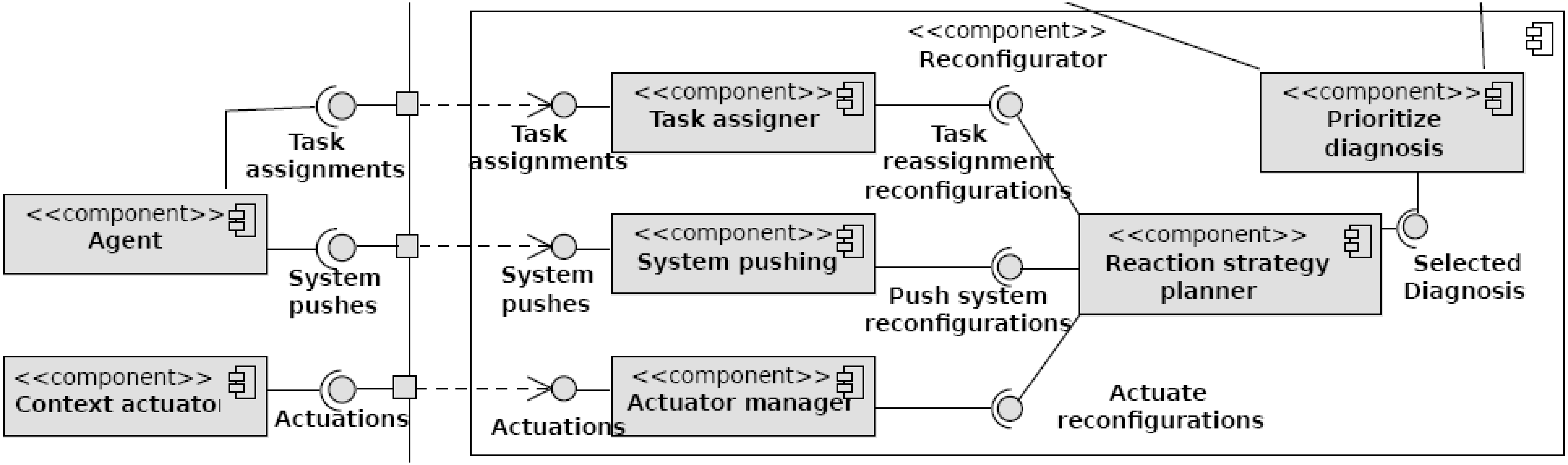
- Check monitored data against contextual goal models and domain assumptions



- A failure occurs when
  - Something that should happen does not occur
  - Something that should not happen does occur

Reconfigurator component

- Reconfiguration types: assign tasks or push (send reminders) agents, control actuators to effect changes
  - Diagnosis are prioritized
  - Compensation actions to revert effects of failed plans



Feedback loops based on requirements awareness

Contact: Vitor E. Souza Silva
(vitorsouza@disi.unitn.it)

Copyright by *Profound Whatever* (taken from Flickr)

Adaptive software systems

- Change their behavior at run-time in response to changes in their environment
- Adaptation mechanism = feedback loop
  - Should fulfill purpose (= requirements)
  - When output indicates otherwise, adapt
  - Must be aware of requirements success/failure
- Which are the requirements that lead to such feedback loop?

Awareness Requirements (AwReqs)

- Refer to other requirements (goals, tasks, quality constraints, domain assumptions) and their success/failure
- Examples:
  - Quality constraint Q should never fail
  - Goal G should complete within 2 hours (delta)
  - Task T shouldn't fail > 3 times a year (aggregate)
  - Failures of domain assumption A won't increase between months (trend)
- AwReqs of AwReqs = Meta-AwReqs.

Adaptivity Requirements

- An AwReq that can also talk about changes of status for another requirement
- Example:

R1 = Requirement R will complete within 2 hours

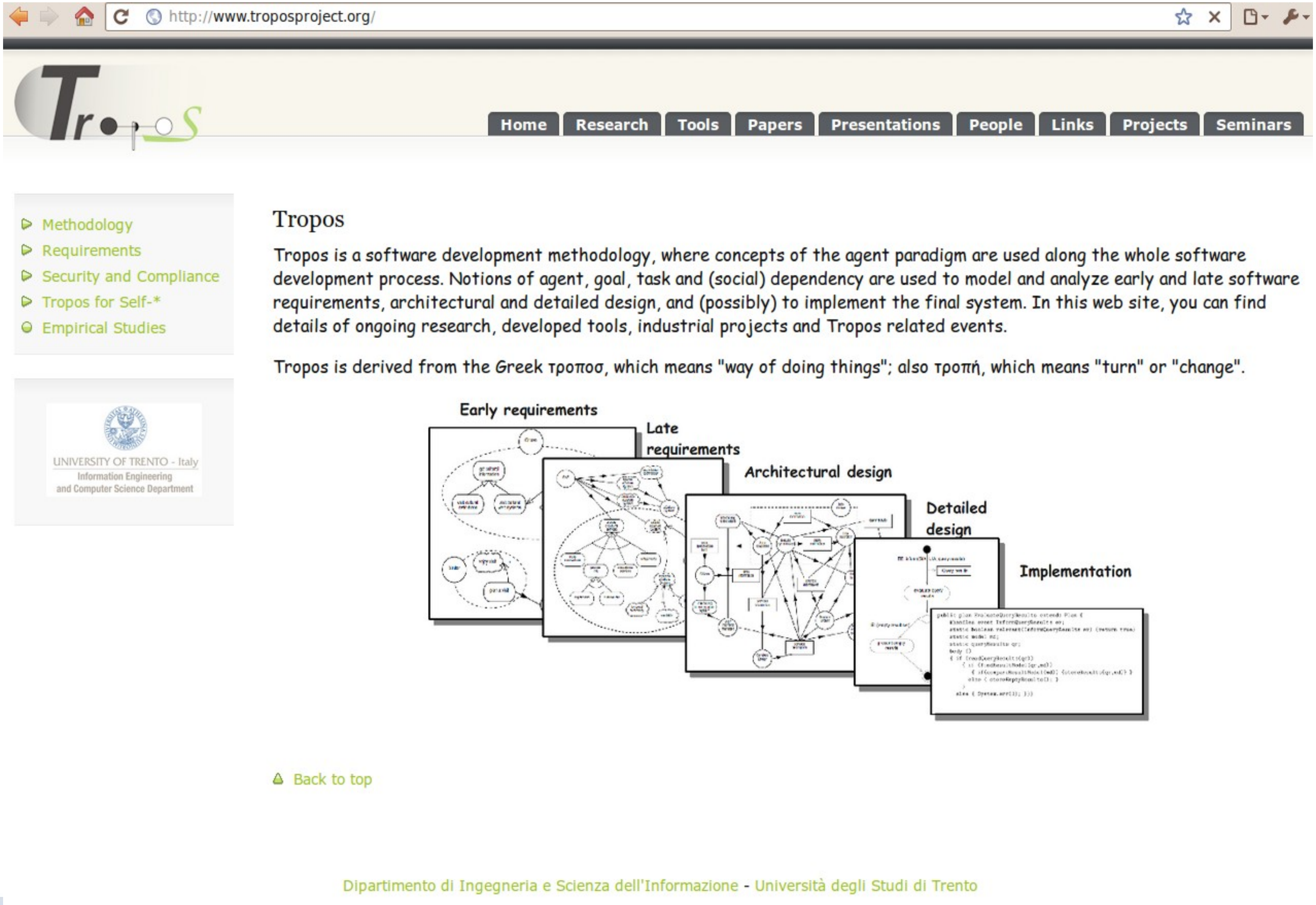
R2 = Requirement R will complete within 3 hours

  - **Relax**: duration(R) > 2h => fail(R1) ∧ initiate(R2)
  - **Good-enough**: 2h < duration(R) < 2.2h => fulfill(R)
  - **Abort**: duration(R) > 4h => fail(R1) ∧ fail(R)
  - **Compensation**: duration(R) > 2h => changeParam(R)

Some research directions

- Contextual security requirements
  - Security requirements (e.g. privacy) can be relaxed sometimes
- Using commitments in adaptive open systems (e.g. STSs)
  - How does one agent adapt in an open system?
  - Which agents to interact with?
- A framework for AwReqs
  - From requirements models to feedback loops
  - Consider contextual requirements

www.troposproject.org



Tropos

Tropos is a software development methodology, where concepts of the agent paradigm are used along the whole software development process. Notions of agent, goal, task and (social) dependency are used to model and analyze early and late software requirements, architectural and detailed design, and (possibly) to implement the final system. In this web site, you can find details of ongoing research, developed tools, industrial projects and Tropos related events.

Tropos is derived from the Greek τροποσ, which means "way of doing things"; also τροπή, which means "turn" or "change".

Dipartimento di Ingegneria e Scienza dell'Informazione - Università degli Studi di Trento