*Modeling with I*²*

Yanik Ngoko

IME-USP

2011

Summary

REQUIREMENT ENGINEERING

In a software development process, Requirement Engineering is the part that produces at the end definition/specification of requirements.

Wishes
Certitudes
Intentions
Requirements proposal
etc.

Formal requirements(
Uses cases, Flowchart,
class diagram etc.)
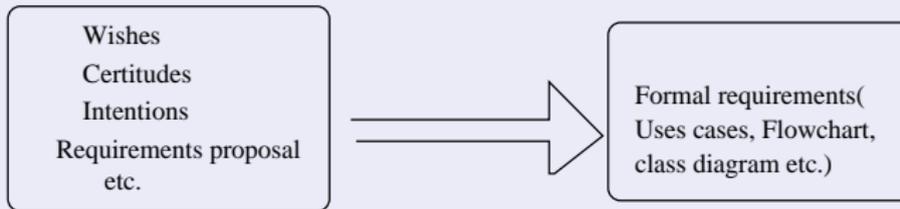
REQUIREMENT ENGINEERING

System context in software development

I* main motivation

Most of requirements modeling techniques (uses cases, Entity association etc.) do not focus on the definition of the system context.

The comprehension of system context is important for many issues

- We have to let clear what the users really want (resources constraint, usage constraints etc.)
- Users of the future system may have many alternatives systems in mind.
- We may have to automate "a chain of work" between workers of an enterprise.

I*

I*

I* is a language for requirements Engineering proposed by Eric Yu and tackling the modelling of system context (or the embedded system in which will be included the system to develop). [a]

a. I* comes before uses cases.

I* USAGES

I* has been used on many types of engineering systems with workflows (health care, air plane transportation, meeting planification, loan service [a] project organization etc.)

a. There exist mapping from BPEL to I*

Technical approach

Intentions

The key concept on I* is actor intentions. Actors are all elements (human, material) whose actions are relevant for the system to develop. I* provides a typology of intentions (task, goal, softgoal etc.). The name "I*" means distributed intentions.

Intention representation

Intentions can be expressed through a relation with another actor (this will give the Strategic Dependency model) or as internal to actor (this gives the Strategic Rational model) [a]

---
a. The difference with Entity Relation is on the typology of intentions

STRATEGIC DEPENDENCY MODEL

Meeting scheduling (view without the meeting scheduler)
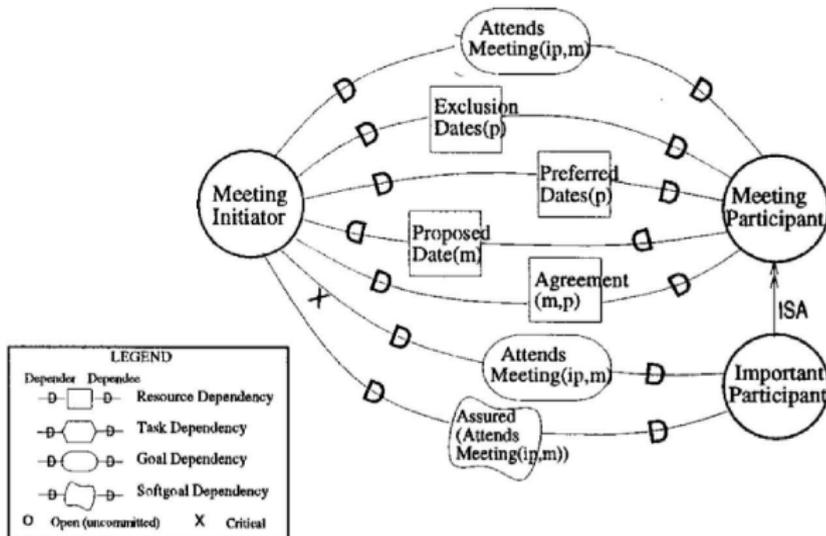
STRATEGIC DEPENDENCY MODEL

Meeting scheduling (view with the meeting scheduler)
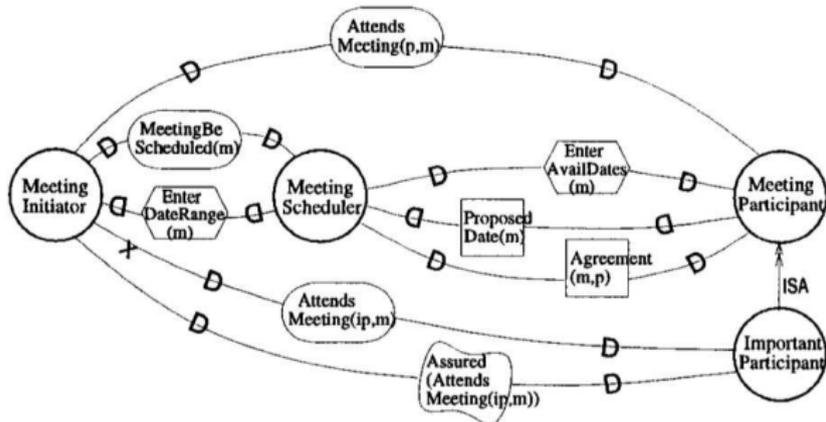
View with internal actors intentions
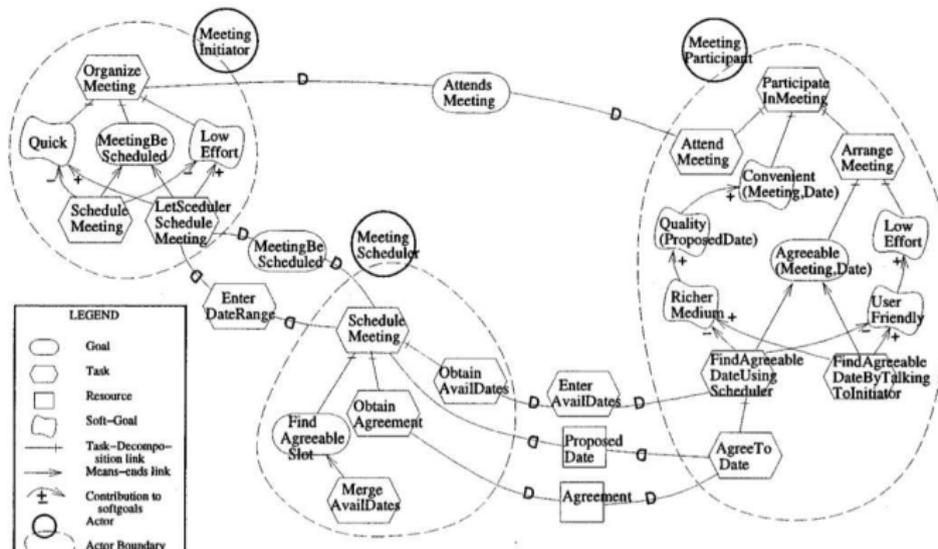
ANALYSIS WITH I*

Routines of the future systems can be analyzed using I* on the following aspect :

- Worbability : Is there a least one actor responsible of each "actions" required for the routine ?
- Viability : Is it workable with all softgoal asserted ?

One can also analyze with I* :

- Ability : For seeing what an actor can do ?
- Believability : Can we believe in ... ?

ANALYSIS PRINCIPLE

The graph representation of "rationale" with I* is an advantage.
For analysing any rationale, we can first :

- Identify the subgraph on which the rationale is expressed ?
- verifiy properties on it

For example a task is workable if all elements to which it is linked
are workable. This can be evaluated by computing the transitive
closure of this task [a].

---

a. Even probabilistic analysis may be envisioned

📄 Eric Yu, *Social Modeling and I**, Research paper, 2003.

📄 Eric Yu, *Towards Modelling an Reasoning Support for Early-Phase Requirements Engineering*, pp 226-235, 1997.

📄 Eric Yu and John Myopoulos, *Understanding "Why" in Software Process Modelling, Analysis and Design*, IEEE Conf on Software Engineering, pp 159-160, 1994.

📄 L. Liu and E. Yu, *Designing Information Systems in Social Context : A Goal and Scenario Modelling Approach*. Information Systems, 29(2) 187-203. Elsevier (2004)