Requirements Engineering in the Days of Social Computing

John Mylopoulos
University of Trento

ICWE'10, Vienna
July 7-9, 2010

Abstract

Thanks largely to Web and other technologies, we are experiencing the rise of a new paradigm for computing that often goes under the label of "social computing". In this paradigm, computing is conducted through services offered by one agent (the server) to another (the client). These services are assembled dynamically and adapt, depending on circumstances. Moreover, the notion of "system" is extended to include software as well as human and organizational agents working together towards the fulfillment of stakeholder requirements. Most importantly, social computing leverages knowledge of human/organizational agents to conduct "computations" that go beyond traditional notions. Early examples of this kind of computing include collaborative filtering, online auctions, prediction markets, reputation systems, etc.

The advent of this paradigm has changed drastically the nature of software requirements. We review traditional and goal-oriented approaches to requirements engineering and argue for the need to extend such approaches (i) to accommodate the modeling and analysis of requirements preferences and priorities, (ii) to accommodate the notion of *social commitment* as the basic building block for specifying solutions to social problems, (iii) to include a new class of requirements we call awareness requirements. Such requirements impose constraints on adaptation mechanisms needed to meet stakeholder needs.

The research reported in this presentation is based on on-going work between the author and Alex Borgida, Amit Chopra, Fabiano Dalpiaz, Neil Ernst, Paolo Giorgini, Ivan Jureta, Alexei Lapouchnian, and Vitor Souza.

Social Computing

- (Weak sense) Refers to systems that support social behaviour, e.g., blogs, social network services, wikis, social bookmarking, …
- (Strong sense) Refers to "computations" that are carried out by groups of human and organizational agents in collaboration with software; examples include collaborative filtering, online auctions, prediction markets, reputation systems, Wikipedia, …
- Has become popular/fashionable because of its relationship to a number of recent trends: (i) social software and Web 2.0, (ii) social networks, (iii) open source as a viable method of software production.

Socio-Technical Systems[+] (STSs)

- These are the systems that conduct social computations.
- They consist of human, software and organizational agents who work together to fulfill stakeholder requirements.
- They are founded on Web and other technologies.
- Agents are inherently autonomous and heterogeneous, and operating environments are unpredictable.
- In order to survive and succeed within such settings, STSs have to be open, dynamic and adaptive.

[+ The term 'socio-technical system' has been used since the 50s in Management Science [Trist51] to refer to systems where human/social concerns are given equal status to technical ones; however, the *nature* of STSs has changed dramatically since the advent of the Web and ubiquitous connectivity]

STSs: How do we *design* them?

- There have been proposals since the '80s [Baxter10] that view the problem as one of work design [Mumford83], vs information system design [Taylor82].
- We focus here on *requirements engineering* (RE) for STSs, as opposed to design (architectural and/or detailed).

Requirements Engineering (RE) for STSs

- There are new types of "typical" functional requirements, e.g., recommendation functions.
- There are also new types of "typical" non-functional requirements, e.g., transparency (→ trust), adaptivity (→ criticality)
- But, is this all?

Basic thesis of this talk: We need new concepts to conduct requirements engineering for STSs.

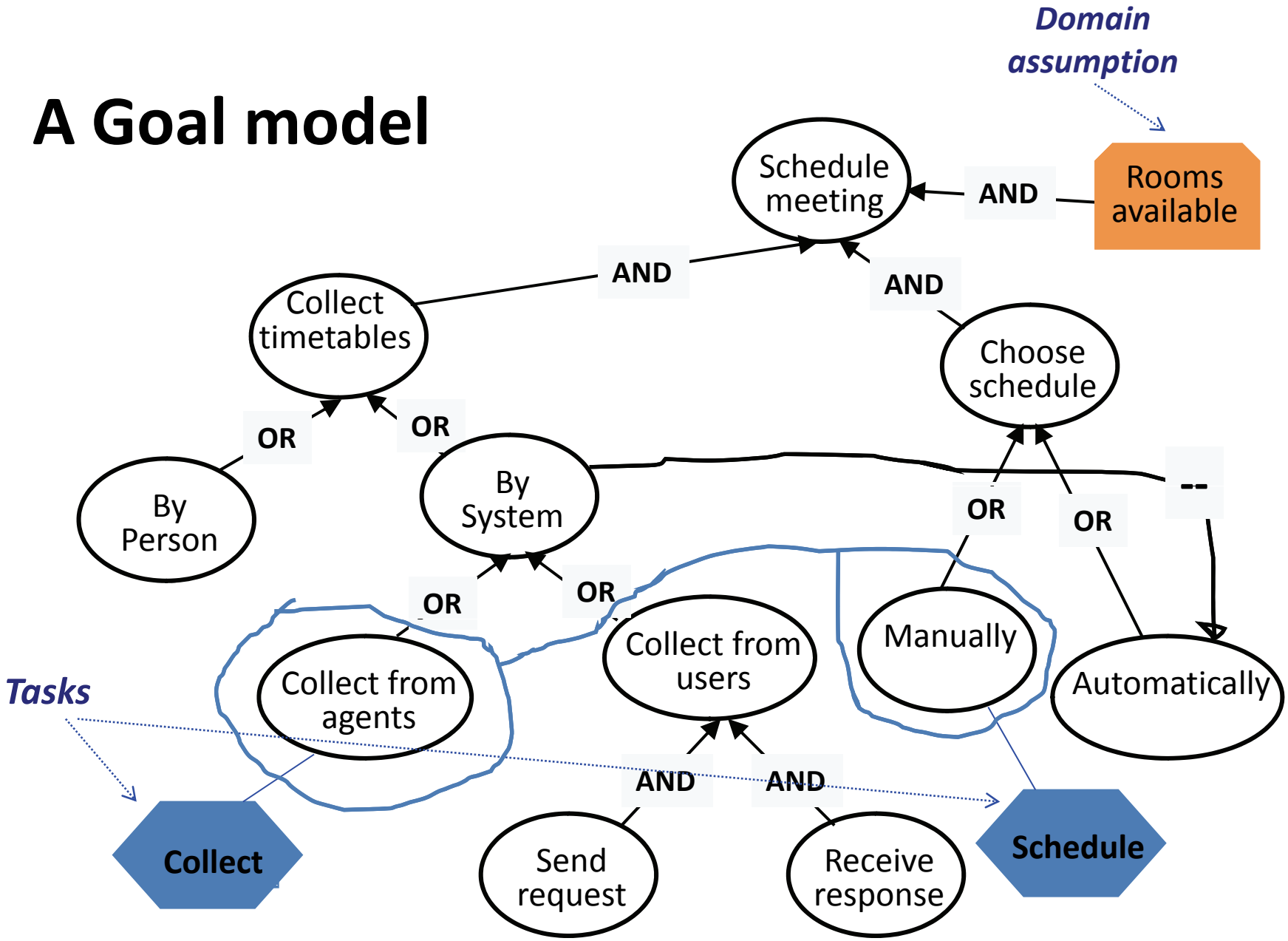
RE: The State-of-the-art in three Ideas

- Requirements are stakeholder goals [KAOS93].
- Non-functional requirements are undefinable goals (softgoals) [NFR92]
- Social settings can be modelled in terms of agents and social dependencies among them (social dependency models) [iStar97].

Requirements as Goals

- Requirements define what stakeholders want (e.g., “schedule meetings”), not what functions the system ought to support (e.g., “request timetables”).
- Goals are refined through AND/OR decompositions until they can be ***operationalized*** through a function/task.
- Goals may be synergistic or contradictory.
- A solution (***specification***) to a given set of goals consists of a bunch of tasks which, if carried out in some order fulfill the goals.
- A goal model defines a space of alternative designs for fulfilling a goal.

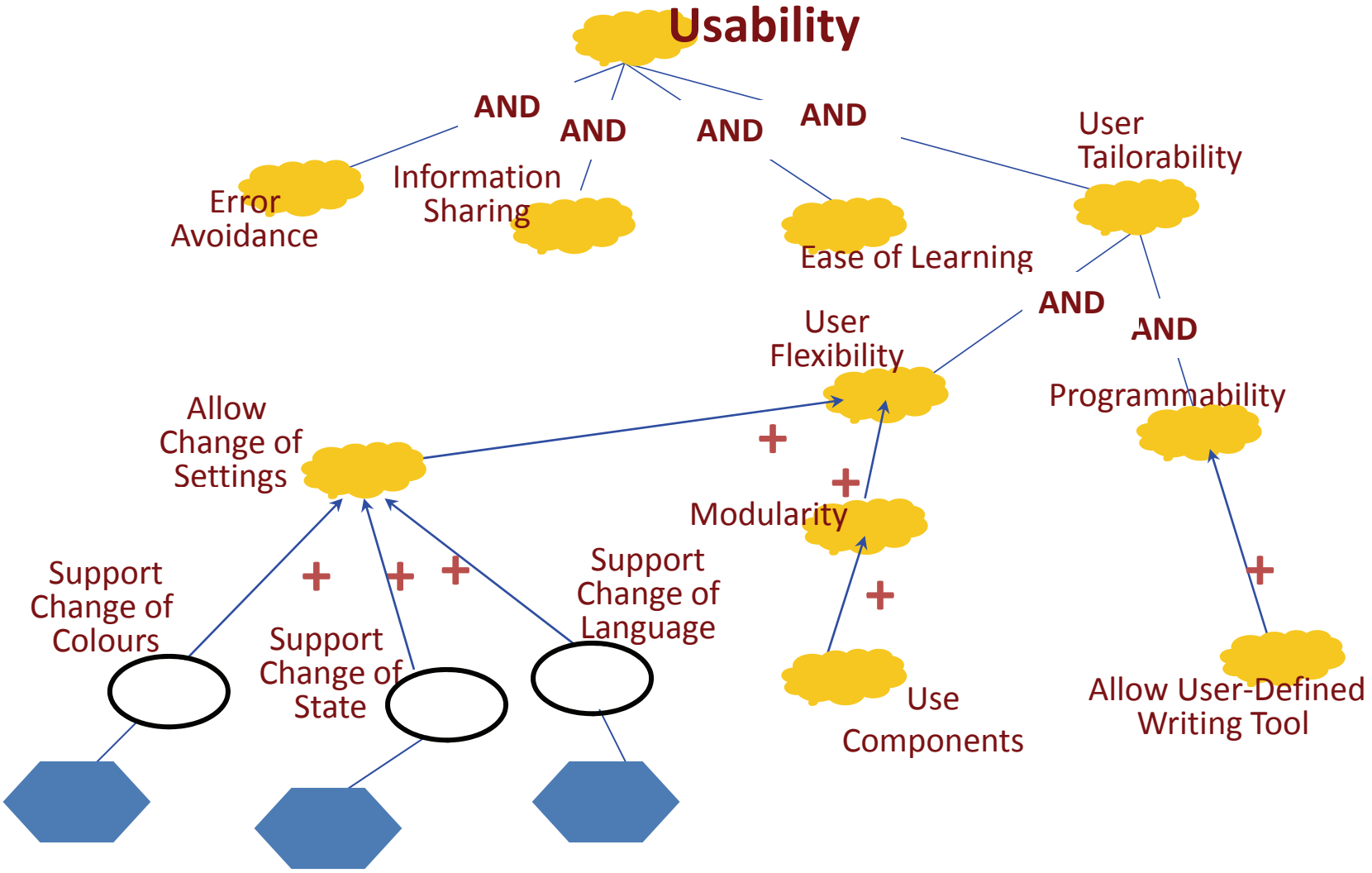
A Goal model


Domain assumption
Schedule meeting
AND
Rooms available
AND
AND
Collect timetables
Choose schedule
OR
OR
By Person
By System
OR
OR
OR
OR
--
Collect from agents
Collect from users
Manually
Automatically
Tasks
AND
AND
Collect
Send request
Receive response
Schedule

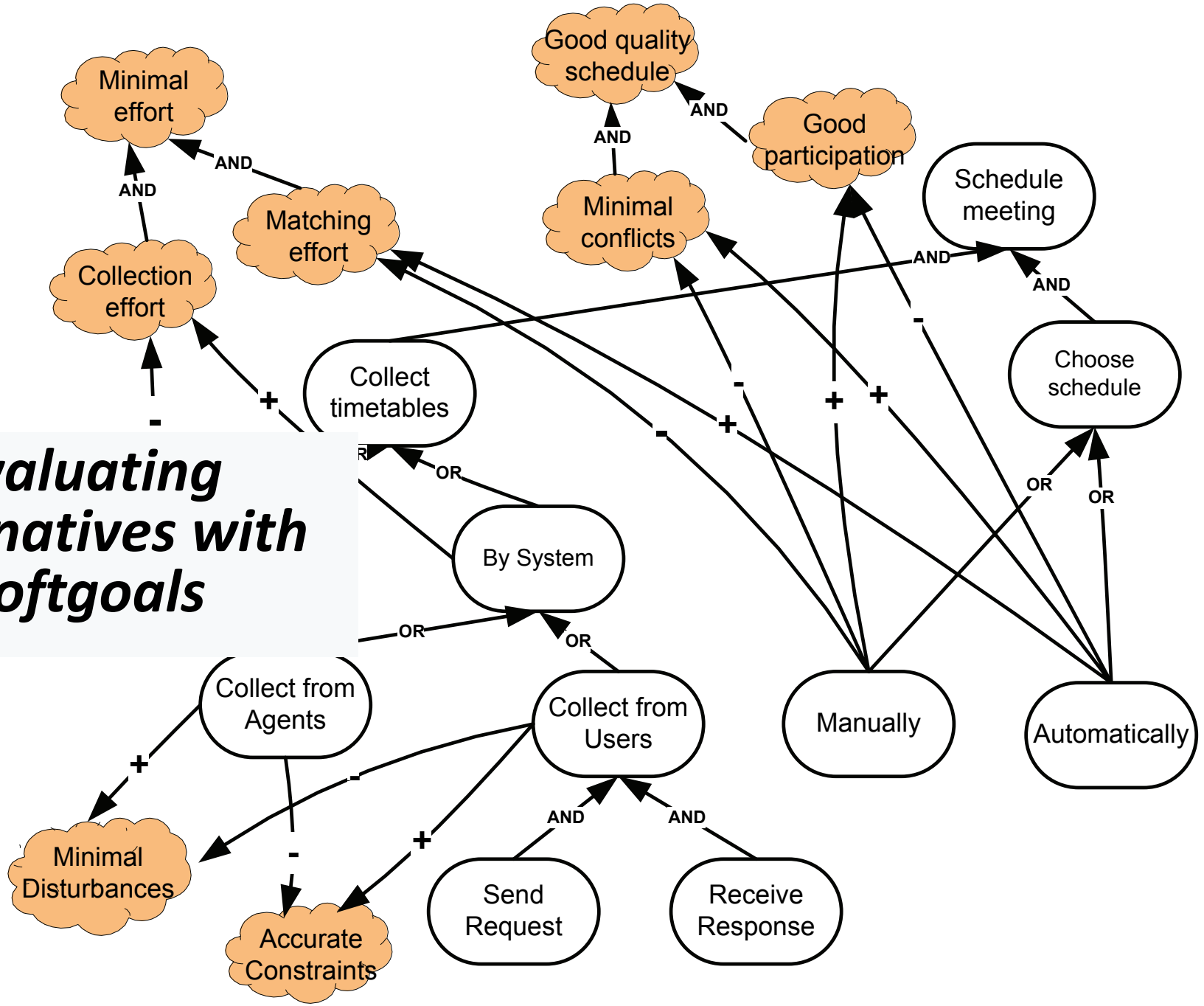
Softgoals

- (Functional) Goals, such as “Schedule meeting” are well defined; non-functional goals, such “higher profits”, “higher customer satisfaction” or “easily maintainable system” specify ***qualities*** a socio-technical system should adhere to and are not definable.
- Such qualities are represented as ***softgoals***; they can be thought as “fuzzy goals” with no clear-cut criteria for satisfaction; hence softgoals are ***satisficed***, rather than satisfied.
- Softgoals can be used as criteria for selecting among alternative designs.

A Softgoal Model



Evaluating Alternatives with Softgoals



Stakeholders and Their Goals

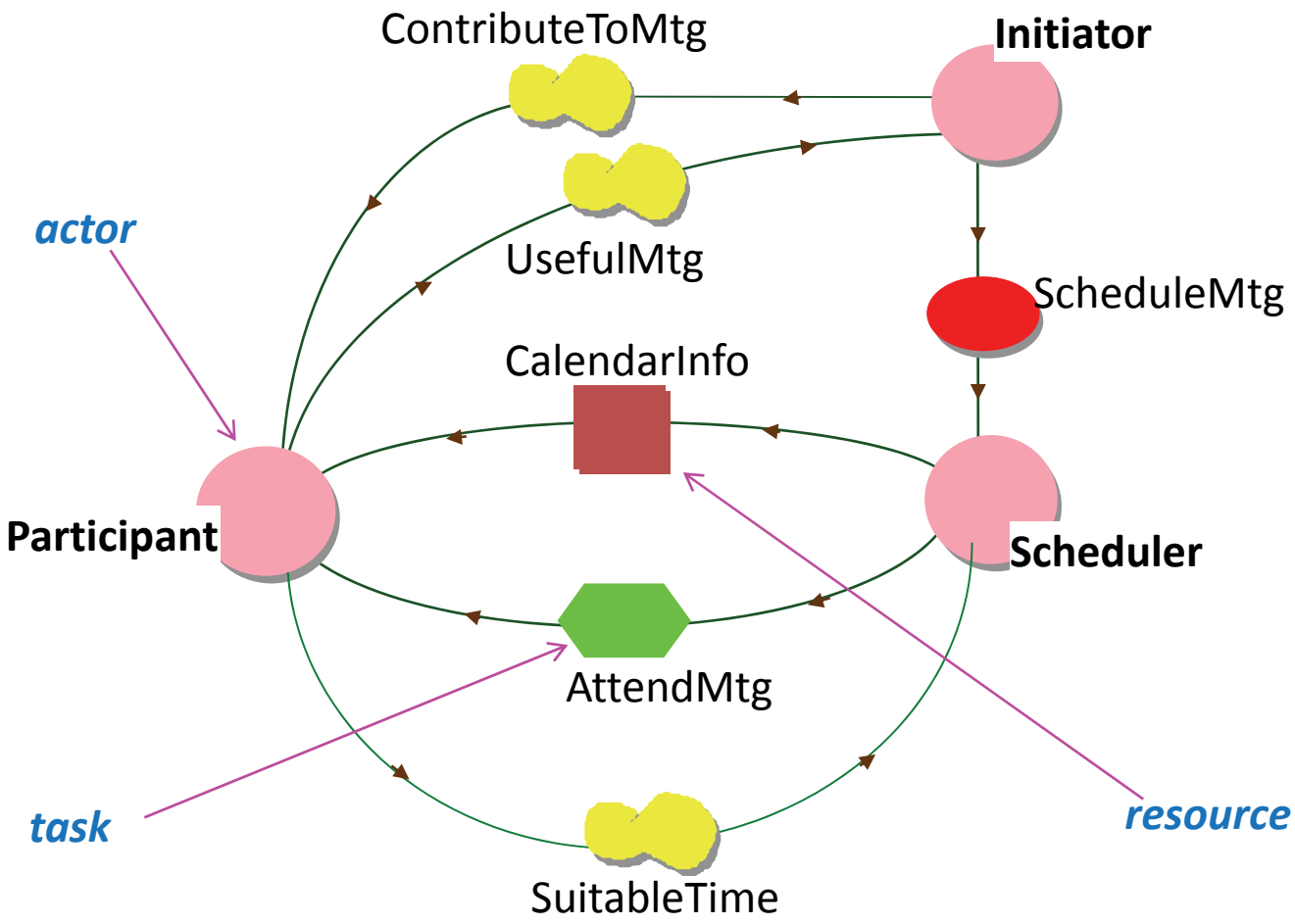
→ In KAOS, goals are global objectives for the system-to-be.

→ In *i** [iStar97], goals are desired by ***actors*** (agents, for our purposes) and are ***delegated*** to other actors for fulfillment.

→ In this framework then, early requirements involve identifying stakeholders and their goals, analyzing these goals, delegating them to other actors etc.

→ The result of this process consists of ***actor dependency*** and ***actor rationale*** models.

An Actor Dependency Model



So, what is missing? …

- Social commitments as primitive building blocks for STS specifications [Chopra10].
- Optional requirements and prioritizations among requirements [Jureta10] (also [Ernst10], [Liaskos10]).
- Awareness requirements for adaptive STSs [Lapouchnian10].

(Social) Commitments

- [Bratman87] and [Cohen90] formalized the notion of an agent's *(psychological) commitment* to her intentions.
- [Singh91] stressed instead the notion of *social commitment* C(a, b, ϕ, ψ) whereby "agent a commits to fulfill ϕ for agent b in return for ψ".
- Think of social commitment as the basic molecule out of which social structures and norms are defined, e.g., obligations to others, allegiance to one's country, to one's employer, to family and friends, …

Commitments and Specifications

- In a social setting, it seems useful to replace the notion of task with that of commitment; after all, designers need to know not only what tasks have to be performed, but also who does what.
- Commitments seem a perfect fit for specifying composite services in that they reflect the intentional+social nature of a service.
- Commitments also offer less operational language for business processes that BP modeling languages.

Commitments vs Actor Dependencies

Actor dependencies in i* have the same flavour as commitments, but there are important differences:

- i* only assumes one-way commitments, i.e., actor a commits to actor b to fulfill ϕ; social commitments are bi-directional: actor a commits to do something for actor b if b commits to do something for a".
- There is a logic of commitments worked out through entailment or inference. For example,

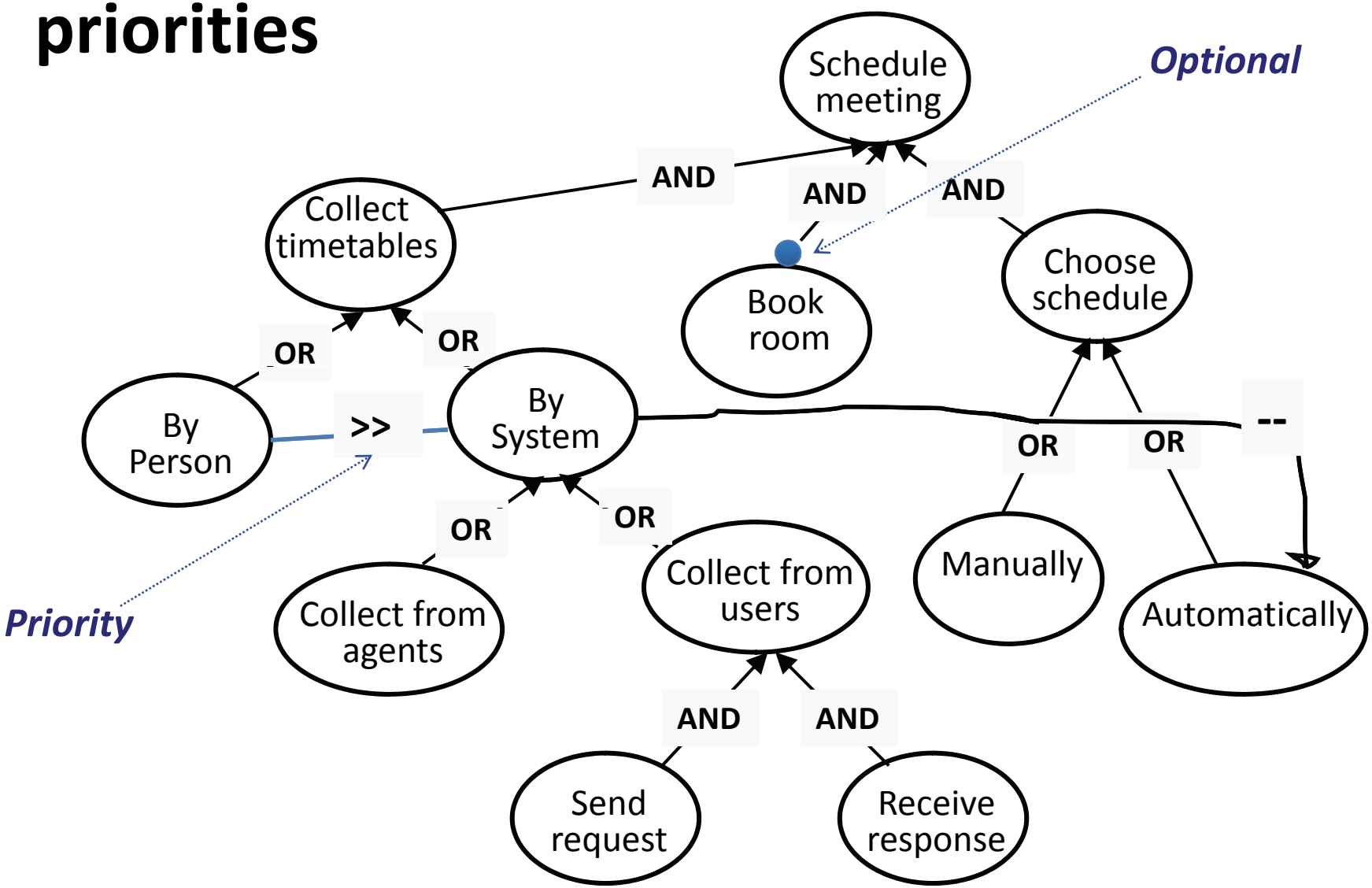
$$C(a, b, \phi_1, \psi), C(a, b, \phi_2, \psi) \models C(a, b, \phi_1 \wedge \phi_2, \psi)$$

- There is also a basic set of speech acts for creating/canceling commitments.

Preferences and Priorities

- In a social setting, taking into account preferences (optional requirements) and priorities makes the difference between a good solution and a non-solution.
- Think of a meeting scheduling service that takes into account preferences such as “Would be nice to also book a room”, “Collecting timetables manually is better than having the system do it”.
- Trouble is: the goal modeling framework presented so far doesn’t allow for representing either preferences or priorities …

A goal model, with preferences and priorities



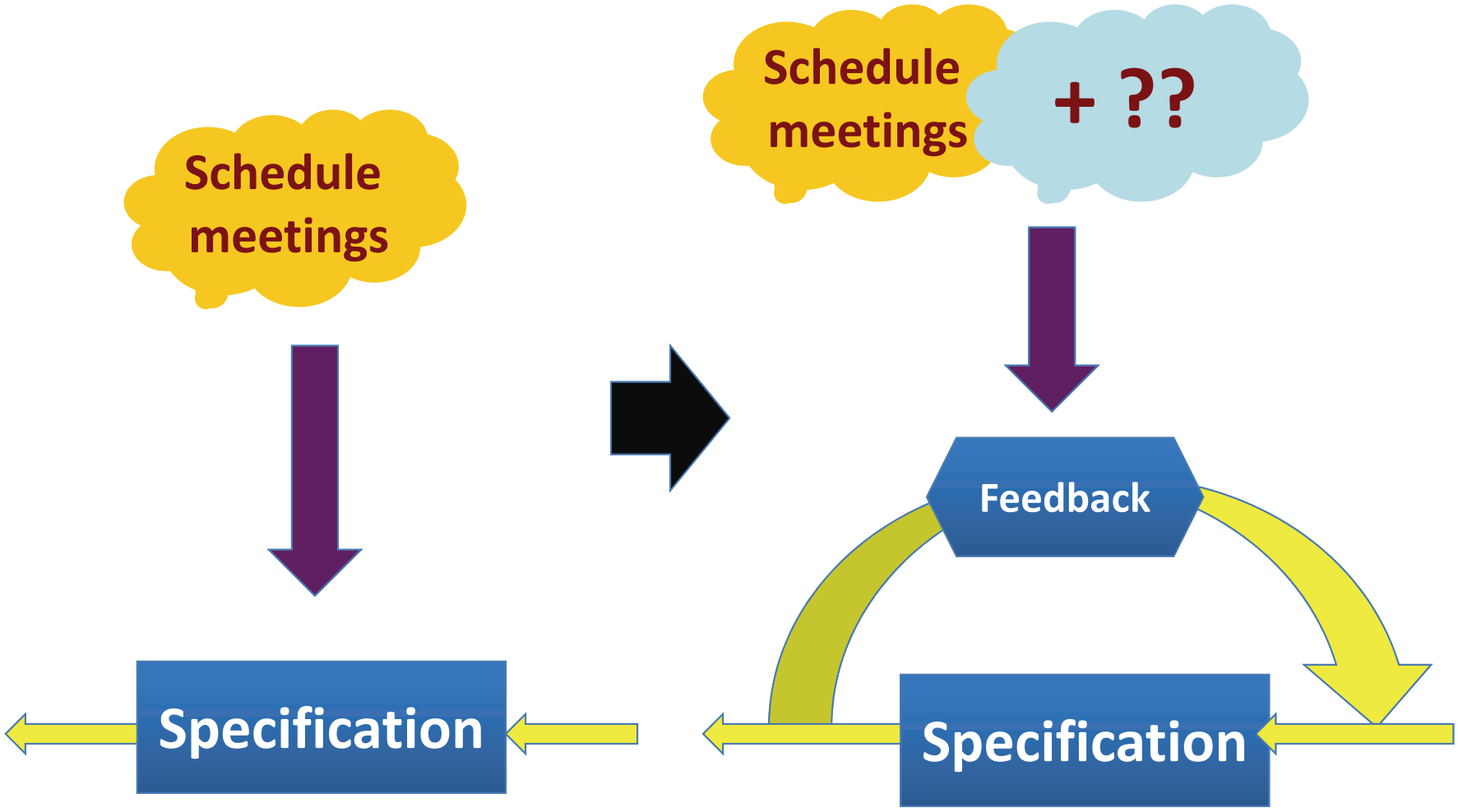
Reasoning with Preferences & Priorities

- Finding a solution to a goal model is now harder: We are looking for solutions that satisfy all mandatory goals, and are maximal wrt preferences and priorities, i.e., satisfy a maximal set of preferences and do best wrt priorities.
- Naive algorithms for finding solutions here are clearly (doubly) intractable. We are exploring heuristic algorithms that come up with good approximations to optimal solutions [Ernst10].

Adaptive systems for social computing

- ***Any*** system – biological, physical, social or computational -- that operates within an uncertain environment needs adaptation mechanisms to survive.
- Adaptation means that the system monitors its operation and the environment and changes configuration/behaviour when things are not working out as planned.
- But, what is to be monitored and what to adapt to? We need a class of requirements that can be operationalized into monitor-diagnose-reconcile-compensate functions.

Requirements for adaptive systems



Awareness

- Awareness: Consciousness, sentience, ability to sense and respond to the environment.
- Many types of awareness play a role in the design of software systems (security/process/context/location …)
- Our perspective: (i) ***Awareness*** gives rise to the need for feedback; (ii) Model awareness requirements; (iii) Propose a new operationalizion for requirements, specifically tailored to awareness.

Awareness requirements

- Refer to other requirements (Goals/Tasks/Domain Assumptions) and their success/failure.
- Consider

  r = ‘schedule meeting’, da = ‘always rooms available’

  r1 = ‘r will be completed within 2hrs’ (delta)

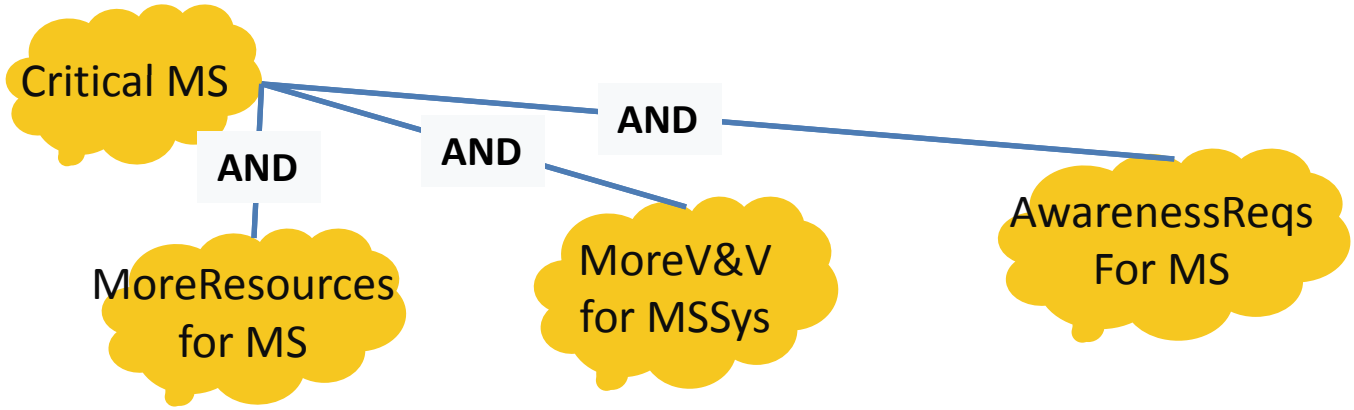
  r2 = ‘r won’t fail >3 times per year’ (aggregate)

  r3 = ‘avg r time won’t increase between months’ (trend)

  r4 = ‘da won’t fail >3 times per year’

Where do awareness reqs come from?

- The need for adaptivity comes from criticality and risk considerations. Criticality, in turn, can have its origins in safety, dependability, reliability, etc.
- Such non-functional requirements constitute the origins of awareness requirements.
- Consider: Meeting scheduling (MS) is a critical requirement for our organization; hence we allocate more resources to MS, we do more V&V for our MS system, AND we impose some awareness requirements for it as well …



Conclusions

- We have argued that the design of socio-technical systems calls for new concepts in terms of which one expresses requirements and new algorithms for finding operationalizations.
- We have noted three areas where extensions to traditional RE concepts are needed. For sure, there are others …
- We are currently exploring these extensions; also working with colleagues from the University of Alicante (Irene Garrigós, Jose-Norberto Mazón and Juan Carlos Trujillo Mondéjar) on the development of an RE framework specifically designed for web engineering.

Epilogue

- The Web has opened the gates to meaningful social interaction for billions of humans.
- But technology is not sufficient on its own for building useful systems for individuals, groups, communities and the wide world.
- To build the STSs of the future, we need to adopt concepts from other disciplines -- notably Philosophy, CogSci, Management Science and Economics – and integrate these into the concepts, tools and techniques that we use for web engineering.

References

[Baxter10] Baxter G., Sommerville I., "Socio-technical systems: From design methods to systems engineering", (draft report).

[Bratman87] Bratman M., *Intention, Plans, and Practical Reason*, Harvard University Press, Cambridge, MA., 1987.

[Chopra10] Chopra, A., Mylopoulos, J., Dalpiaz, F., Giorgini, P., Singh, M., "Requirements as Goals and Commitments too", in Salinesi, C., Souveyet, C., Ralyte, J. (eds.) *Intentional Perspectives on Information Systems Engineering*, Studies in Computational Intelligence book series, Springer-Verlag, June 2010.

[Cohen90] Cohen P., Levesque H., "Intention is choice with commitment", *Artificial Intelligence 42*, 1990, 213–261.

[Ernst10] Ernst N., Borgida A., Jureta I., Mylopoulos J., "Reasoning with Optional and Preferred Requirements", 29th International Conference on Conceptual Modeling (ER'10), November 2010.

[iStar97] Yu E., "Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering", 3rd IEEE Int. Conference on Requirements, Engineering, 1997, 226-235.

[Jureta10] Jureta, I., Borgida, A., Ernst, N., Mylopoulos, J., "Techne: Towards a New Generation of Requirements Modeling Languages with Goals, Preferences and Inconsistency Handling", 19th Int. IEEE Conference on Requirements Engineering (RE'10), September 2010.

References (cont'd)

[KAOS93] Dardenne A., van Lamsweerde A., Fickas S., "Goal-directed Requirements Acquisition", *Science of Computer Programming 20*, 1993, 3-50.

[Lapouchnian10] Lapouchnian A., Souza V., Mylopoulos J., "Awareness Requirements for Adaptive Systems", (unpublished document).

[Liaskos10] Liaskos, S., McIlraith, S., Mylopoulos, J., "Integrating Preferences into Goal Models for Requirements Engineering", 19th International IEEE Conference on Requirements Engineering (RE'10), September 2010.

[Mumford83] Mumford, E. "Designing human systems for new technology - The ETHICS method", retrieved from http://www.enid.eu-net.com/C1book1.htm

[NFR92] Mylopoulos, J., Chung, L. and Nixon, B., "Representing and Using Non-Functional Requirements: A Process-Oriented Approach," *IEEE Transactions on Software Engineering 18(6)*, June 1992, 483-497.

[Singh91] Singh, M., "Social and psychological commitments in multi-agent systems", AAAI Fall Symp. on Knowledge and Action at Social and Organizational Levels, 1991,104–106.

[Taylor82] Taylor, J., "Designing an Organization and an Information-System for Central Stores – a Study in Participative Socio-Technical Analysis and Design", *Systems Objectives Solutions 2(2),* 1982, 67-76.

[Trist51] Trist, E., Bamforth, K., "Some social and psychological consequences of the long-wall method of coal-getting", *Human Relations 4*, 1951, 3-38.