Requirements Engineering for Services

Presented by Neil Ernst

Work with John Mylopoulos, Alex Borgida, Amit Chopra, Fabiano Dalpiaz, Paolo Giorgini, Ivan Jureta, Alexei Lapouchnian, and Vitor Souza

University of Toronto

MISSE, Waterloo

July 22, 2010

Socio-Technical Systems (STS)

- Consist of human, software and organizational agents who work together to fulfill stakeholder requirements.
- Agents are inherently autonomous and heterogeneous, and operating environments are unpredictable and changing.
- In order to survive and succeed within such settings, STS have to be open, dynamic and adaptive.
- Service-orientation is a key enabler of dynamic STS.

Requirements Engineering (RE) for STS

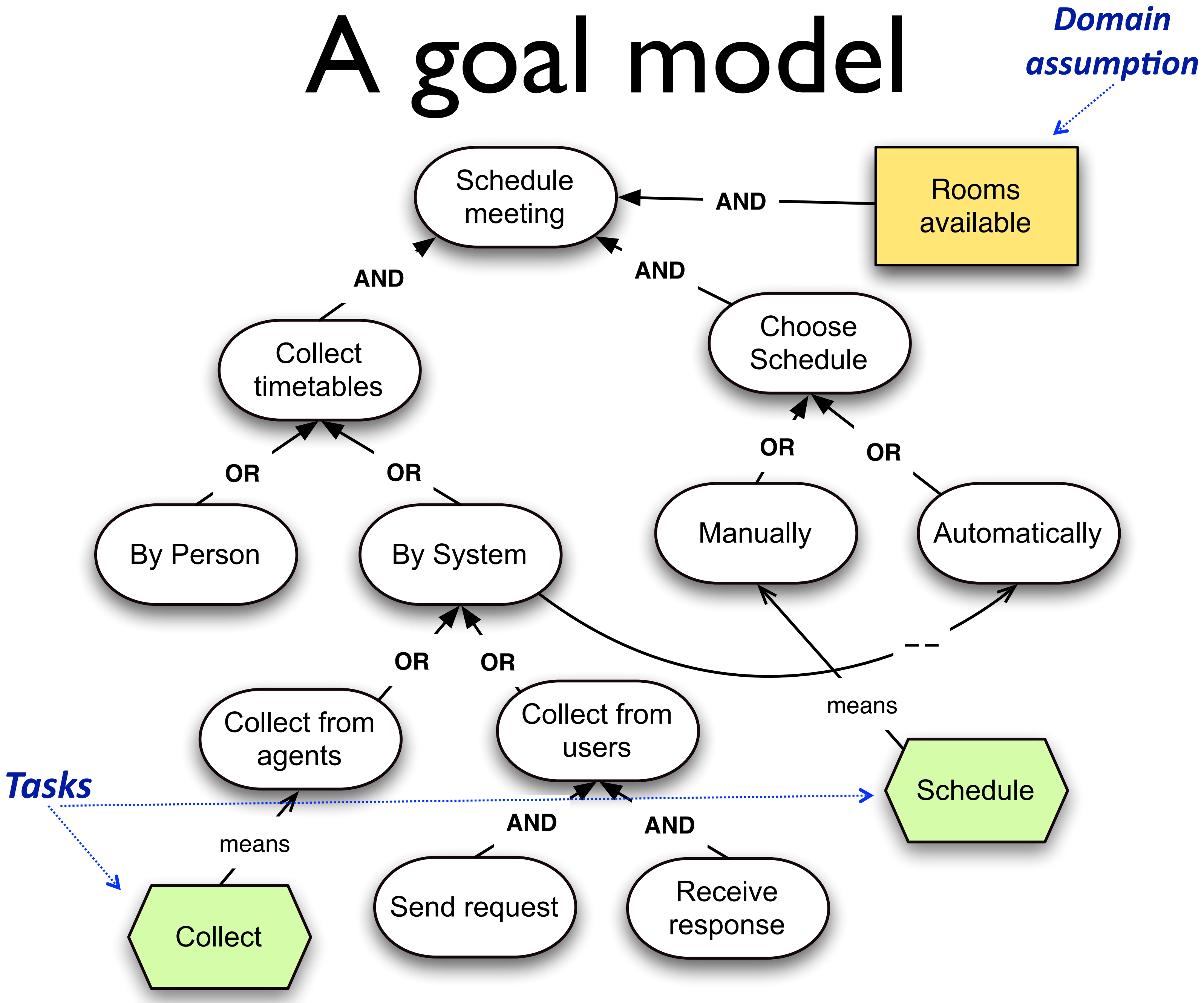
- There are new types of "typical" functional requirements, e.g., recommendation functions.
- There are also new types of "typical" non-functional requirements, e.g., transparency (→ trust), adaptivity (→ criticality)
- But, is this all?

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

Requirements as Goals

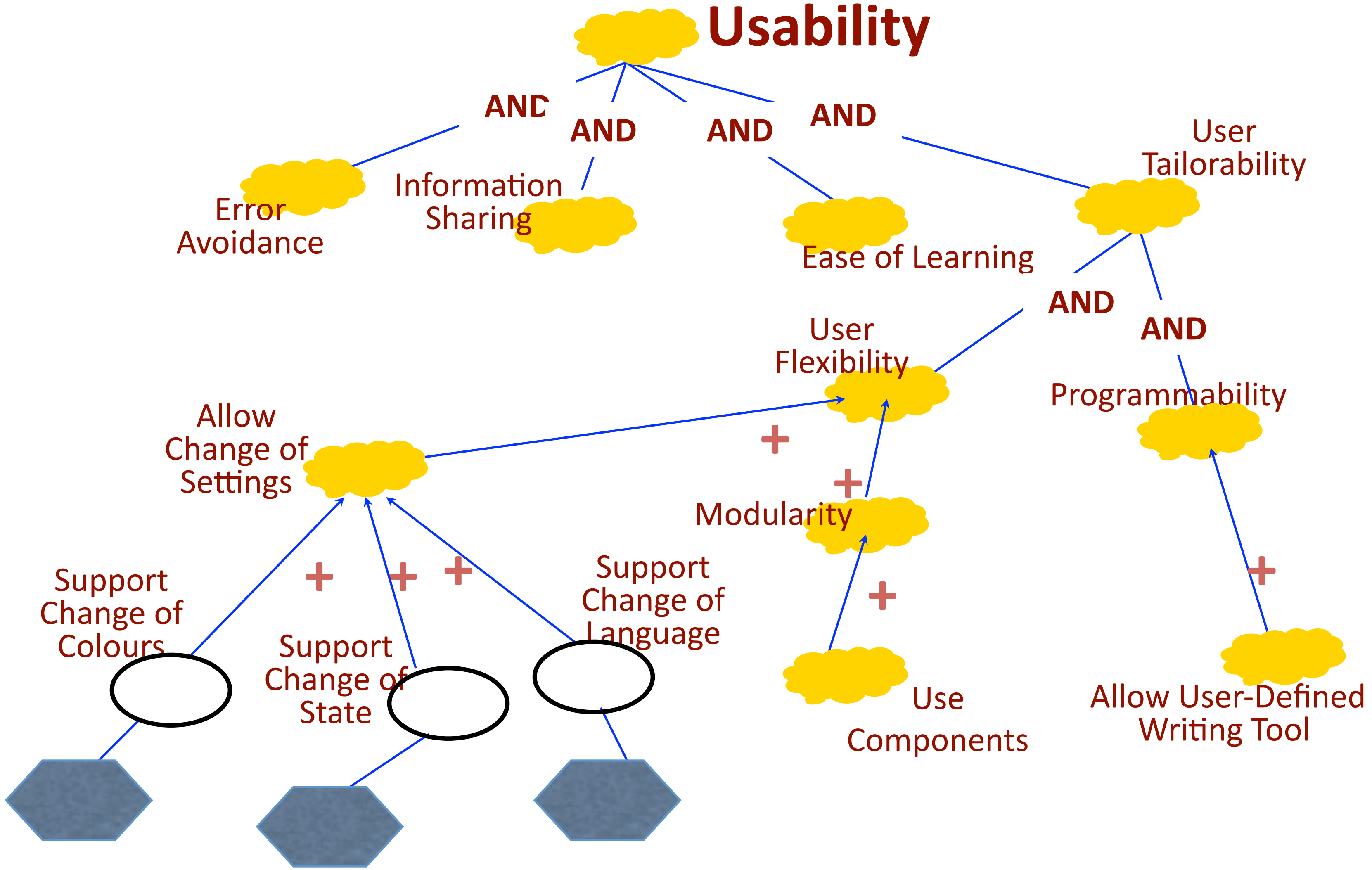
- Requirements define what stakeholders want (e.g., "schedule meetings"), not what functions the system ought to support (e.g., "request timetables").
- Goals are refined through AND/OR decompositions until operationalized by a function/task.
- Goals may be synergistic or contradictory.
- The requirements problem: given a set of goals, which tasks and assumptions (specification) satisfy that set?
- A goal model defines a space of alternative designs for fulfilling a goal.

A goal model

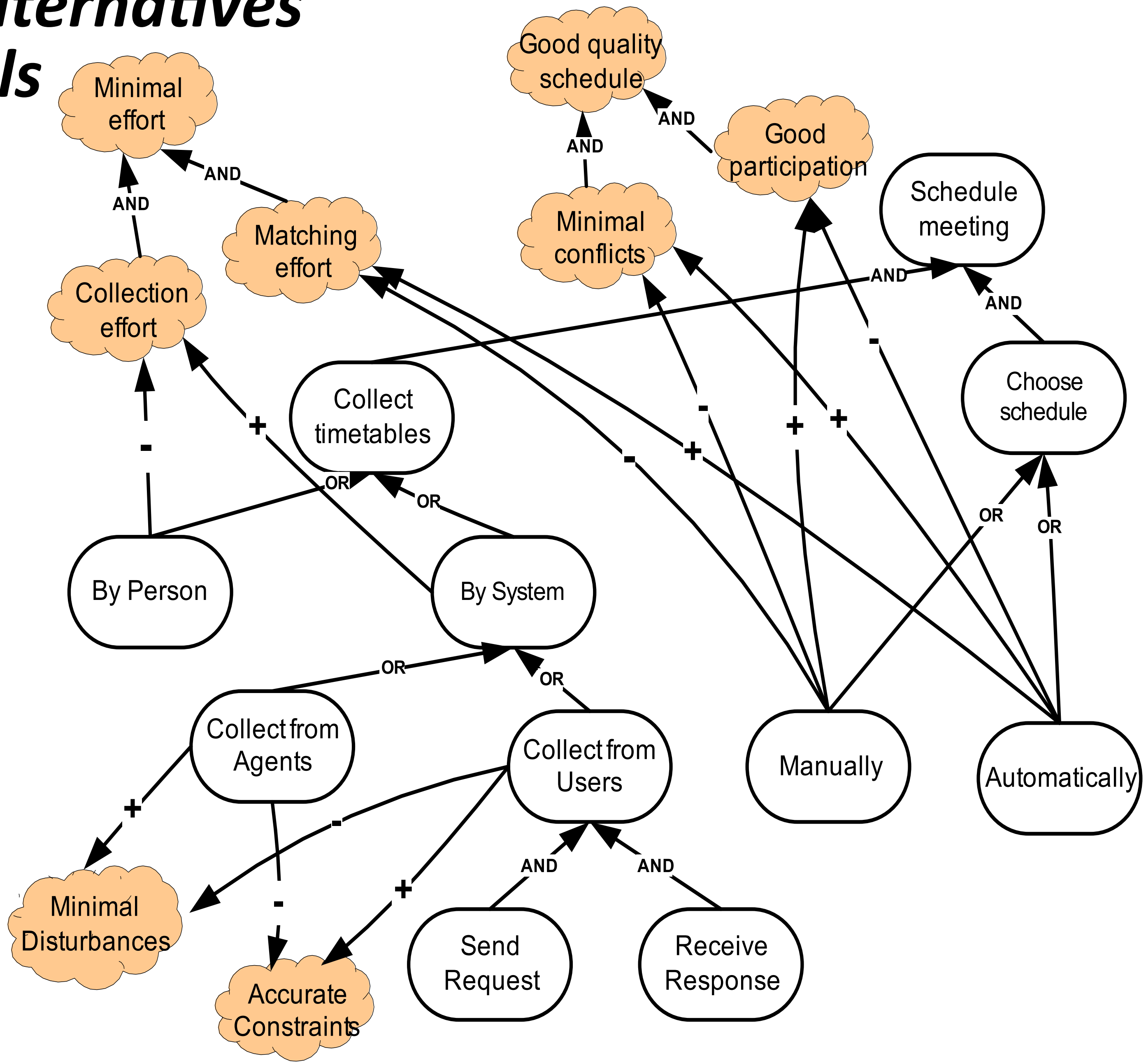


Domain assumption
Schedule meeting
AND
Rooms available
AND
AND
Collect timetables
Choose Schedule
OR
OR
OR
OR
By Person
By System
Manually
Automatically
OR
OR
means
Collect from agents
Collect from users
Schedule
Tasks
AND
AND
means
Collect
Send request
Receive response

A Softgoal Model



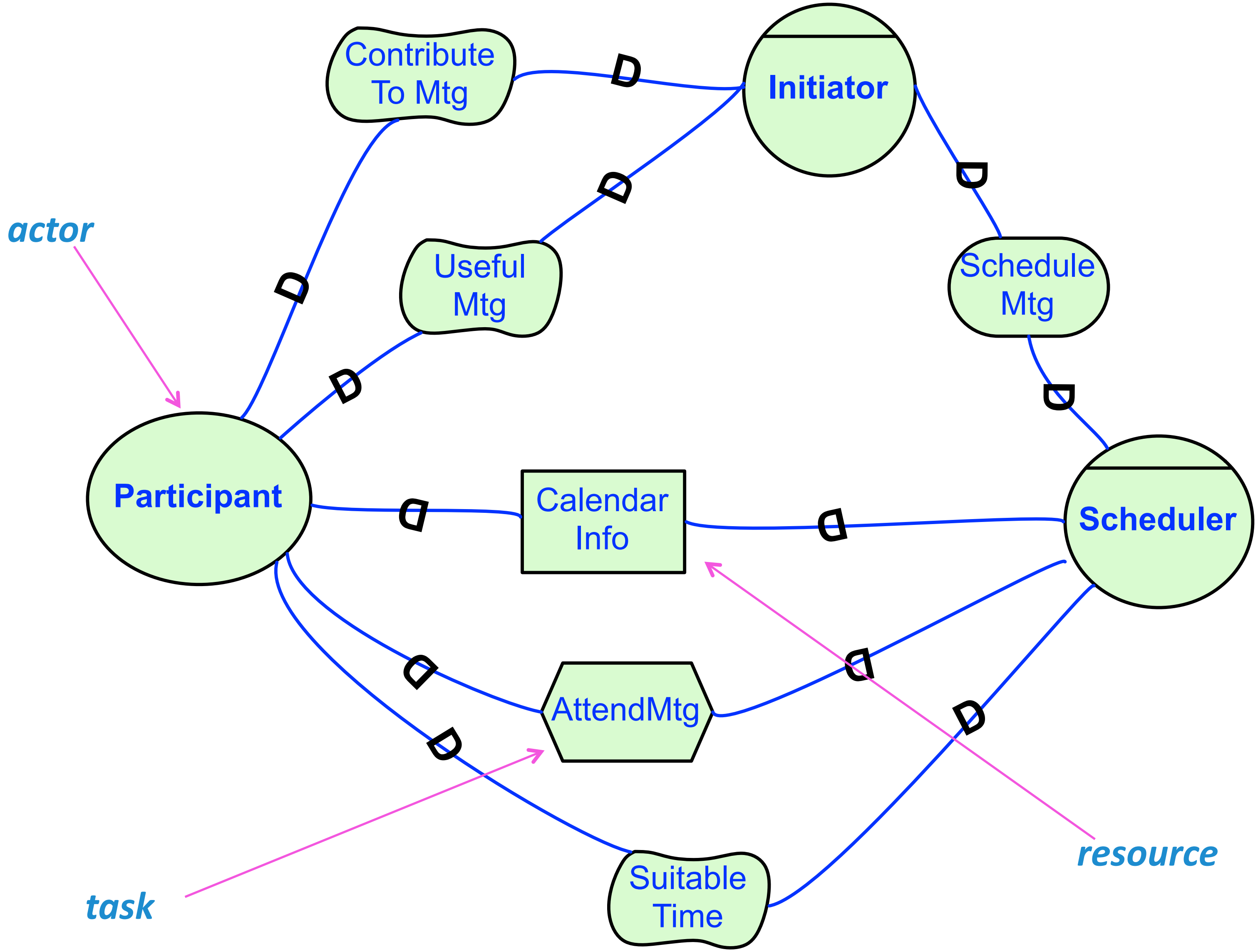
Evaluating Alternatives with Softgoals



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].
- Awareness requirements for adaptive STSs [Lapouchnian10].
- Optional requirements and prioritizations among requirements [Jureta10],[Ernst10], [Liaskos10].

Commitments

(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 ψ for agent b, on condition b does ϕ for a".
- 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 than BP modeling languages.

Awareness requirements

Awareness of ...

- 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.

Awareness

- Many types of awareness play a role in the design of software systems (security/process/context/location …)
- Awareness gives rise to the need for feedback.
- Must model awareness requirements.
- Develop a new operationalization for requirements, specifically tailored to awareness.

Awareness requirements

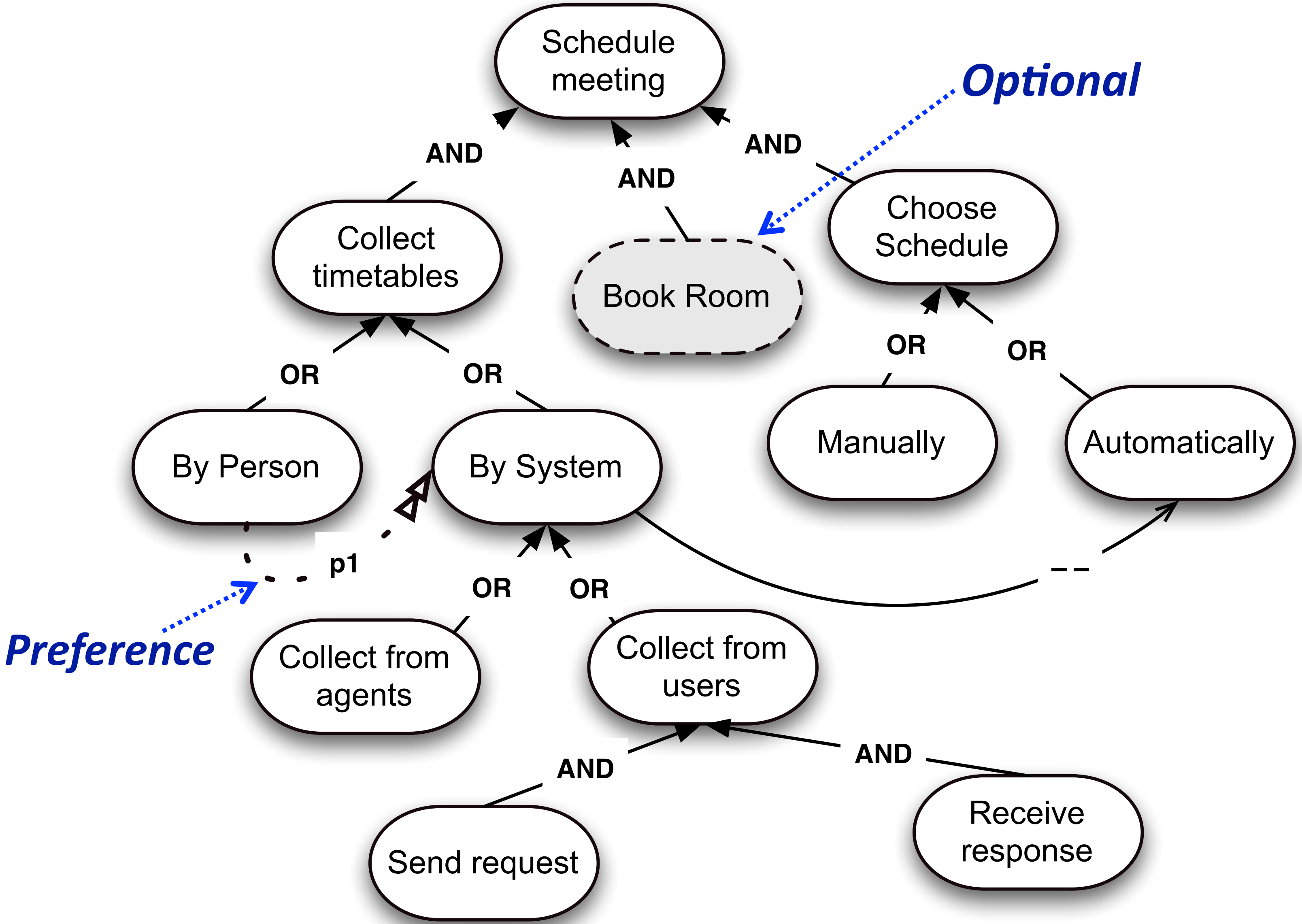
- Refer to other requirements (Goals/Tasks/Domain Assumptions) and their success/failure.
- Consider
  - r = 'schedule meeting', d_a = 'always rooms available'
  - r_1 = 'r will be completed within 2hrs' (delta)
  - r_2 = 'r won't fail >3 times per year' (aggregate)
  - r_3 = 'avg r time won't increase between months' (trend)
  - r_4 = 'd_a won't fail >3 times per year'

Prioritizing requirements

Preferences and Options

- In a social setting, taking into account preferences and optional requirements makes the difference between a good solution and a non-solution.
- There are usually multiple solutions to a given requirements problem.
- 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 …

Prioritizing the goal model



Run-time requirements

- STS are now expected to be always-on and always-relevant
- But: operating environment is changing (see Belady and Lehman)
- System updates should be managed wrt requirements (run-time/reflective requirements) [Qureshi09, Cheng09]
- Models must be analyzed quickly:
  - heuristically
  - incrementally

Reasoning with Preferences & Options

- Introduce a RML with support for new concepts, Techne [Jureta10]
- Finding solutions to req problems is now harder:
  - We are looking for solutions that satisfy all mandatory goals, and are maximal wrt options and preferences, i.e., satisfy a maximal set of preferences and do best wrt options.
- 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].

Finding optimal solutions

- We group elements of a model into consistent sub-nets
- Each subnet is checked for admissibility
- Admissible subnets are compared:
  - how many optional goals included?
  - how many preferences satisfied?
- Naive comparisons are intractable
- Approach: local search of the Pareto-front of solutions

Ranking solutions

- Consider a model with two preferences (p_1, p_2) and two options (o_3, o_4)
- The model has admissible solutions S_a and S_b
- Tabu search finds a solution which is locally optimal wrt {p,o}

	p_1	p_2	o_3	o_4
S_a	no	yes	yes	yes
S_b	yes	no	no	yes

Ranking solutions

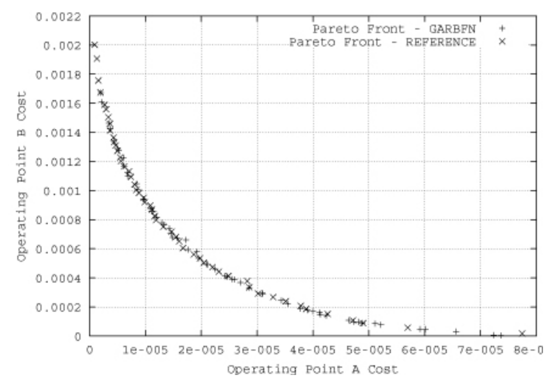
- Consider a model with two preferences (p_1, p_2) and two options (o_3, o_4)
- The model has admissible solutions S_a and S_b
- Tabu search finds a solution which is locally optimal wrt {p,o}

	p_1	p_2	o_3	o_4
→ S_a	no	yes	yes	yes
S_b	yes	no	no	yes

Ranking solutions

- Consider a model with two preferences (p_1, p_2) and two options (o_3, o_4)
- The model has admissible solutions S_a and S_b
- Tabu search finds a solution which is locally optimal wrt {p,o}

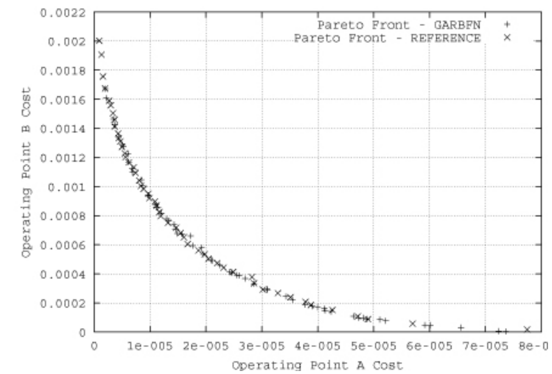
	p_1	p_2	o_3	o_4
→ S_a	no	yes	yes	yes
S_b	yes	no	no	yes



Ranking solutions

- Consider a model with two preferences (p_1, p_2) and two options (o_3, o_4)
- The model has admissible solutions S_a and S_b
- Tabu search finds a solution which is locally optimal wrt {p,o}
- Other decision criteria possible

	p_1	p_2	o_3	o_4
→ S_a	no	yes	yes	yes
S_b	yes	no	no	yes



Conclusion

- We have argued that the design of adaptive, online STS calls for new ways to express 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 …
- There are many lessons from other disciplines – e.g. Philosophy, CogSci, Management Science and Economics – that need to be considered

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.

UNIVERSITY of TORONTO

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.

[Cheng09] B.H. Cheng, R. de Lemos, H. Giese, P. Inverardi, and J. Magee, "Software Engineering for Self-Adaptive Systems: A Research Roadmap," Software Engineering for Self-Adaptive Systems, Springer-Verlag, 2009, pp. 1-26.

[Qureshi09] N.A. Qureshi and A. Perini, "Engineering adaptive requirements," Workshop on Software Engineering for Adaptive and Self-Managing Systems, Vancouver, BC: 2009, pp. 126-131.

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
  - There is a basic set of speech acts for creating/canceling commitments.
  - There is a logic of commitments worked out through entailment or inference. For example,

$$C(a,b,\phi 1,\psi),\ C(a,b,\phi 2,\psi) \models C(a,b,\phi 1 \land \phi 2,\psi)$$