A Concept of Agent for Software Development

Eric Yu
University of Toronto
June 2001

AOIS workshop @ CAiSE'01

What is AOIS? Why AOIS?

- **AOIS is a (specialized) class of information technology solutions ?**
  - compare: software agents technology, data mining tech.,…
  - Solutions looking for problems?
- **AOIS is a (specialized) class of IS applications?**
  - compare: e-commerce, m-commerce, digital libraries,…
  - Problems looking for solutions?
- **AOIS is a new conception of what information systems are, by adopting "agent" as a key abstraction.**
  - Agent concepts can (should) be used throughout conceptualization, requirements analysis, architecture, design, and implementation, as well as during ongoing support.
  - Provides techniques for expressing problems, solutions, and for matching problems with solutions (at each stage)

Problems & Solutions in IS & SE

- **Problems in IS/SE**
  - high costs, frequent failures, …
  - legacy, evolution, …
  - inflexibility, incompatibilities, …
- **Solutions in IS/SE**
  - technologies, platforms, …
    - e.g., C/S, dist'd systems, DBs, web technologies,
  - models, languages, tools
  - formal methods, informal (structured) methodologies for system development – requirements, design, implementation,…
  - reuse, patterns, frameworks, …

Problems & Solutions in AOIS

- **Problems addressed by AOIS ?**
  - high costs, frequent failures, …
  - legacy, evolution, …
  - inflexibility, incompatibilities, …

same as in IS/SE + new opportunities

- **Solutions offered by AOIS ?**
  - agent technologies, platforms, …
  - models, languages, tools
  - methodologies for system development ??
    - – requirements, design, implementation,…
  - reuse, patterns, frameworks, …??

Points to remember

1. **AOIS (and IS) is not just about technology.**
   - Systems exist in a social/organizational environment.
   - ISD/SE continues to be intensely a human activity.

So…

- **If agent concepts are to be used throughout conceptualization, requirements analysis, architecture, design, and implementation, as well as during ongoing support,**
  - **what concept(s) of agent is appropriate?**
  - **what properties/characteristics should it have?**
  - **what abstractions should the agent concept provide?**

Conception of Agent as a Computational Abstraction

e.g., Jennings, Sycara, Wooldridge (1998)

- **Situated**
  - sense the environment and perform actions that change the environment
- **Autonomous**
  - have control over their own actions and internal states
  - can act without direct intervention from humans
- **Flexible**
  - responsive to changes in environment, goal-oriented, opportunistic, take initiatives
- **Social**
  - interact with other artificial agents and humans to complete their tasks and help others

Analysis and Design of Agent-Oriented Systems

e.g., Wooldridge Jennings Kinny (JAAMAS 2000) "GAIA"

- **Analysis level**
  - Roles and Interactions
    - Permissions
    - Responsibilities
      - » liveness properties
      - » safety properties
    - Activities
    - Protocols
- **Design level**
  - Agent types
  - Services
  - Acquaintances

Modelling concepts being driven from programming again?!!

- Structured Analysis from Structured Programming
- OOA from OOD, OOP
- AOA from AOP ??

Requirements Engineering

- **relationship between system and environment.**
  - *Bubenko (1980), Greenspan (1982), Jackson (1983)…*
- **Traditional focus:**
  - consistency, completeness, …
  - e.g., "Three Dimensions of RE" *Pohl (1993)*
    - informal -> formal (representation)
    - opaque -> complete (specification)
    - personal view -> common view (agreement)
- **Recent:**
  - goals, scenarios, agents *van Lamsweerde (ICSE 2000)*

Points to remember

1. **AOIS (and IS) is not just about technology.**
   - Systems exist in a social/organizational environment.
   - ISD/SE continues to be intensely a human activity.
2. **Need to model relationships between machine and world.**

Ontologies for Modelling

- **Static Ontologies**
- **Dynamic Ontologies**
- **Intentional Ontologies**
- **Social Ontologies**

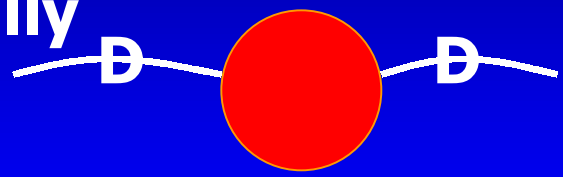
- *Most current conceptions and models of information systems are based on static and dynamic ontologies.*
  - *business process models*
  - *workflow models*
  - *enterprise models*

[J. Mylopoulos CAiSE 97 Keynote]

i - agent-oriented modelling*

I* objectives, premises, key concepts

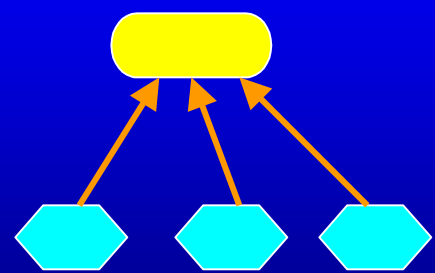
- **Actors are semi-autonomous, partially knowable**



- **Strategic actors, intentional dependencies**

wants and abilities

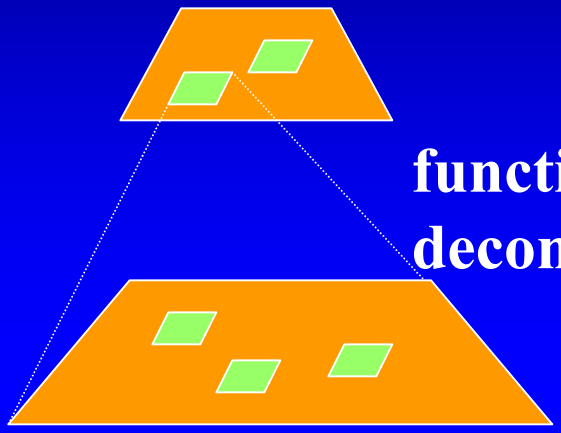
- **have choice, reasons about alternate means to ends**



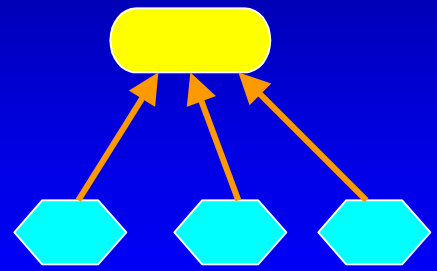
means-ends alternatives

i modeling*

1. explicit intentionality → goals



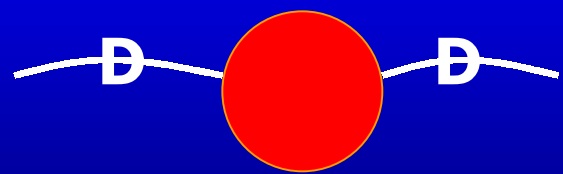
functional
decomposition



means-ends
alternatives

2. implicit intentionality → agents

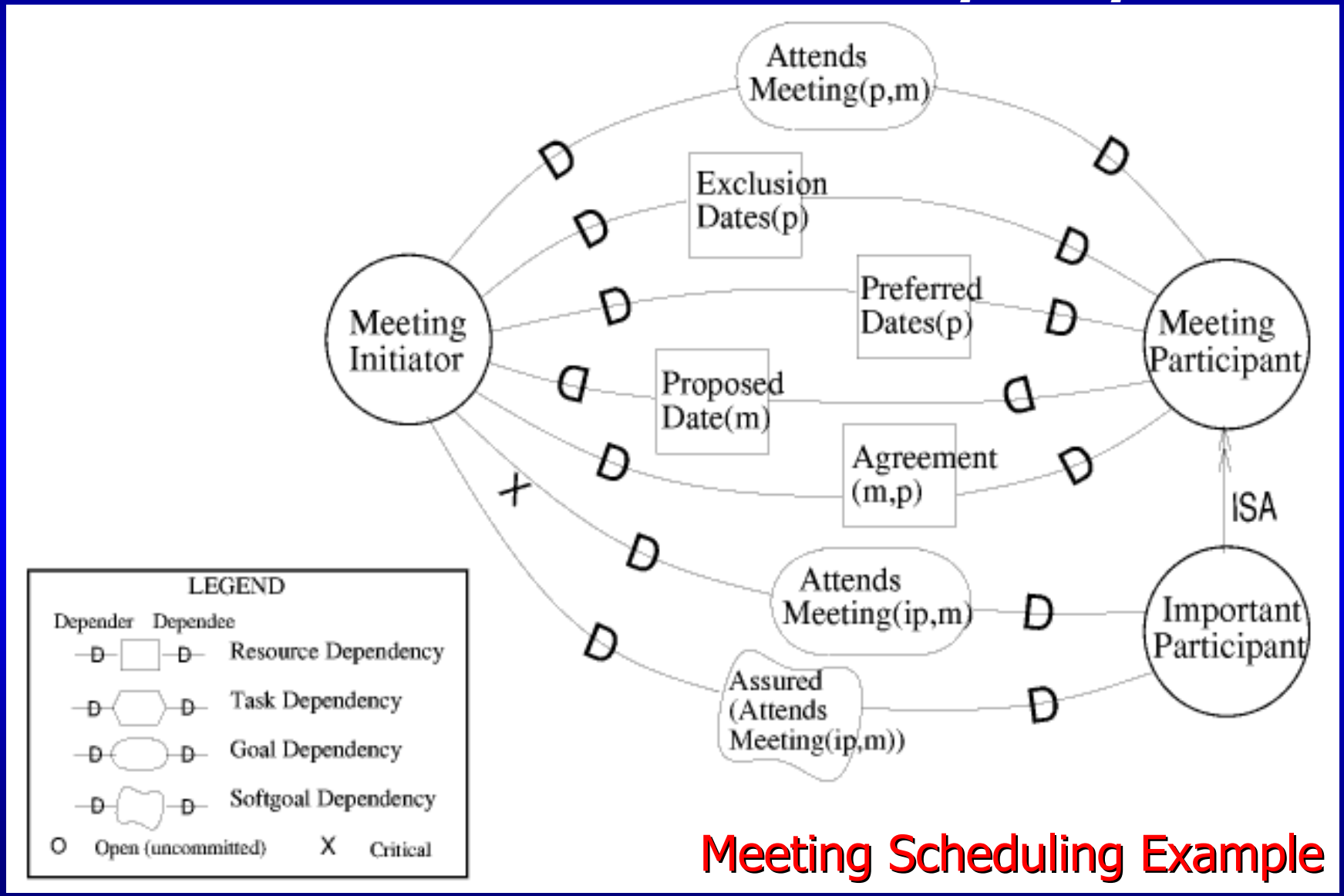
inputs
outputs



wants and
abilities

"Strategic Dependency" Model

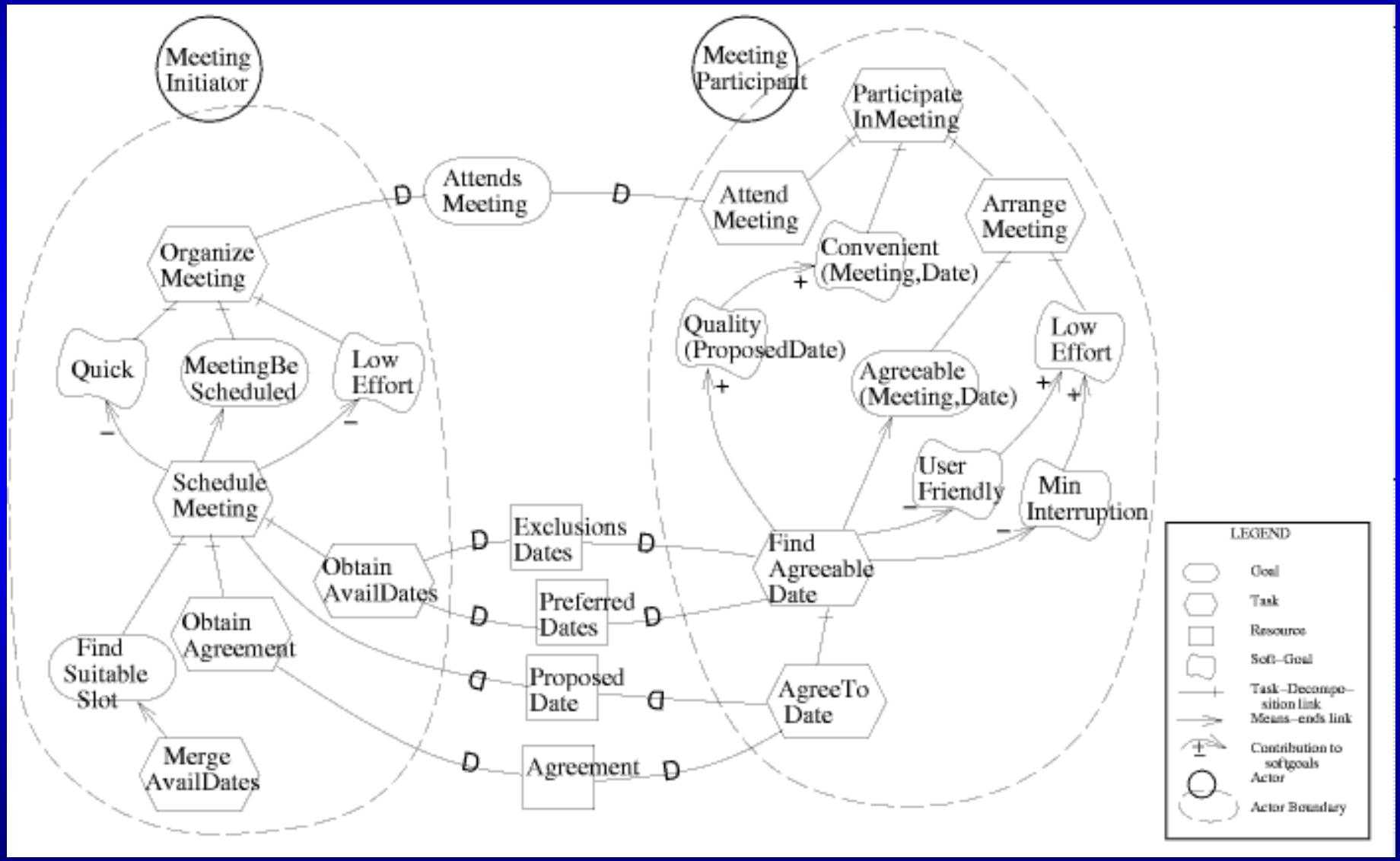
[Yu RE97]

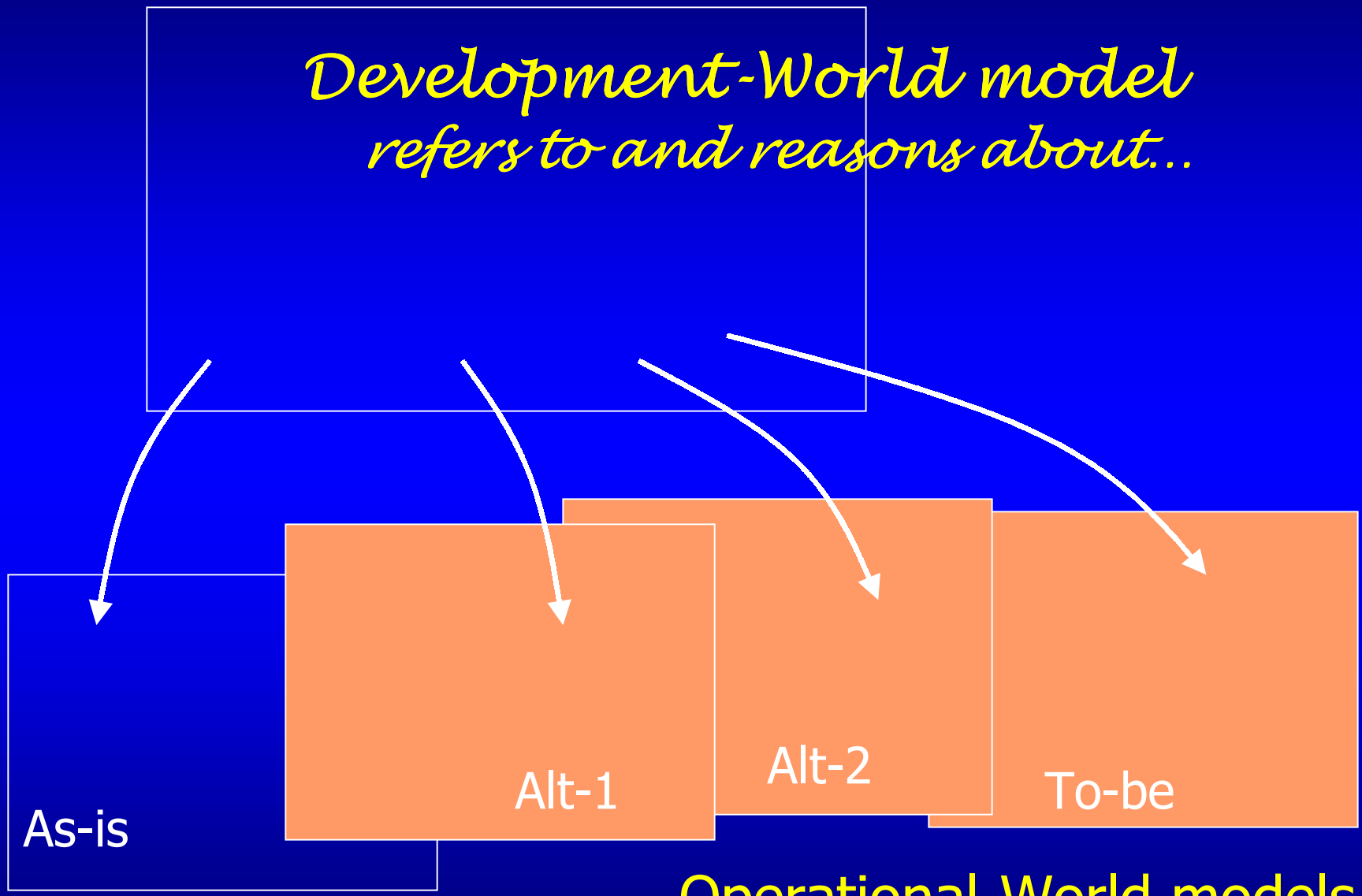


Meeting Scheduling Example

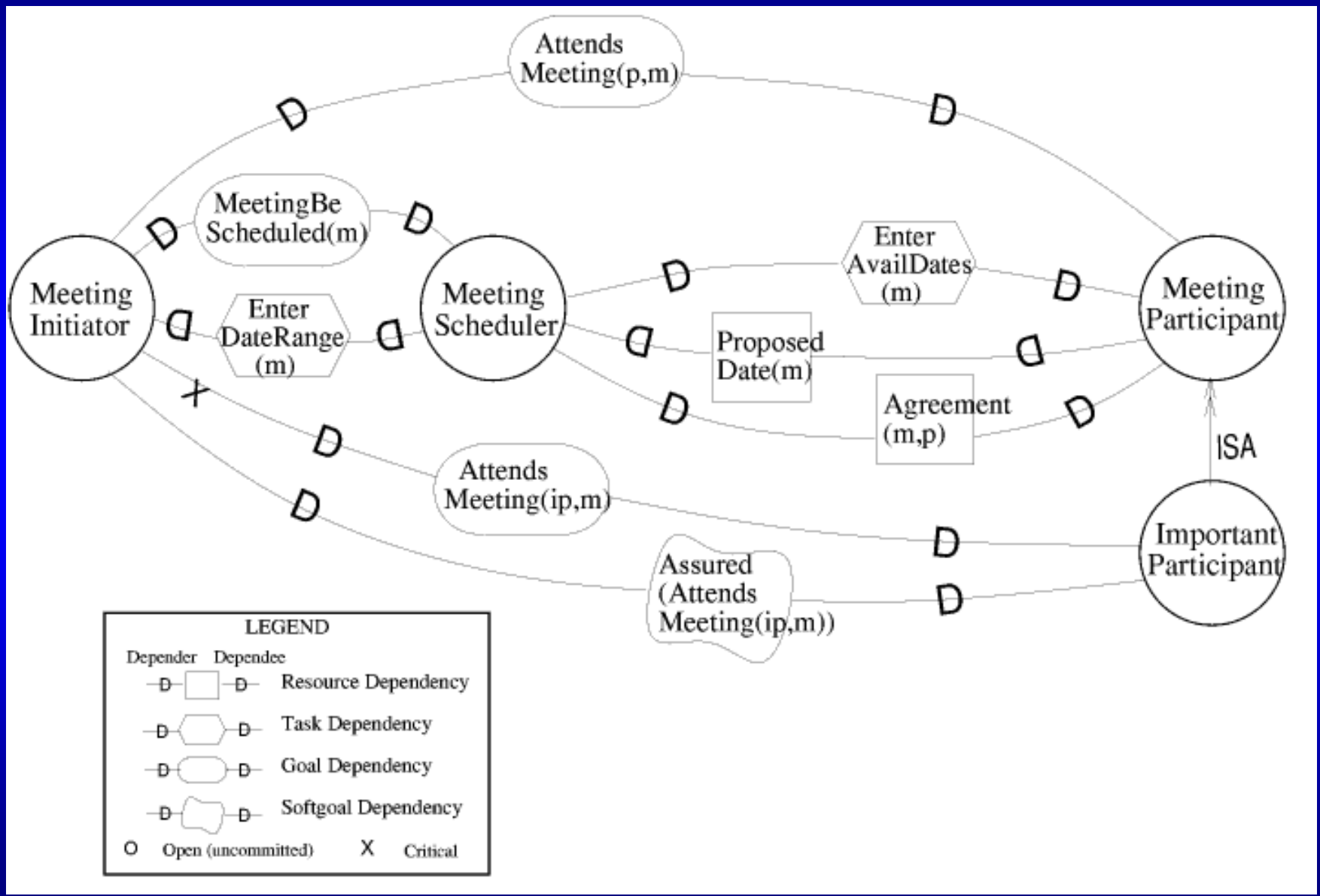
Revealing goals, finding alternatives

- **Ask "Why", "How", "How else"**

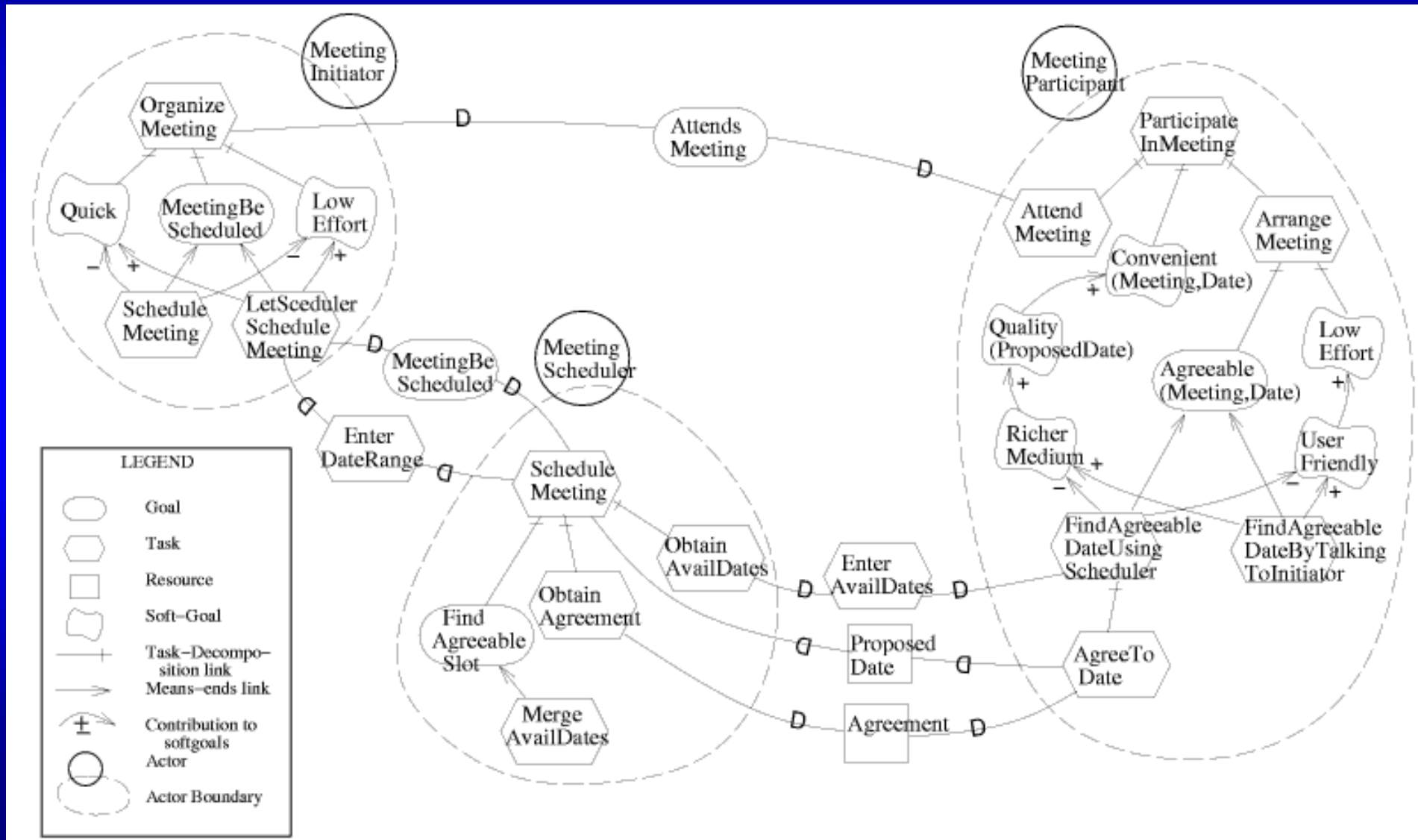



Development-World model
refers to and reasons about..
As-is
Alt-1
Alt-2
To-be
Operational-World models

Scheduling meeting …with meeting scheduler



"Strategic Rationale" Model with Meeting Scheduler



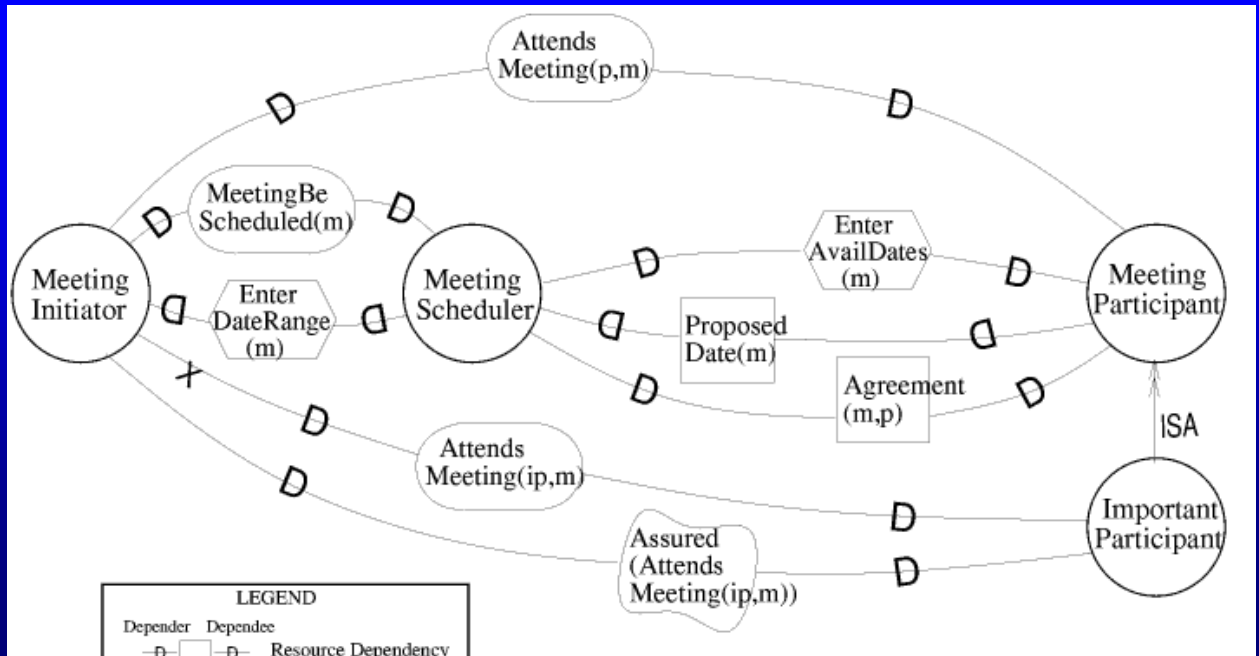
So what are the important concepts for Agent Orientation as a Modelling Paradigm ?

- **Intentionality**
- **Autonomy**
- **Sociality**
- **Identity & Boundaries**
- **Strategic Reflectivity**
- **Rational Self-Interest**

E. Yu. "Agent Orientation as a Modelling Paradigm," Wirtschaftsinformatik, April 2001.

Agent Orientation as a Modelling Paradigm

1. **Intentionality**
   - Agents are intentional.
   - Agent intentionality is externally attributed by the modeller.
   - Agency provides localization of intentionality.
   - Agents can relate to each other at an intentional level.

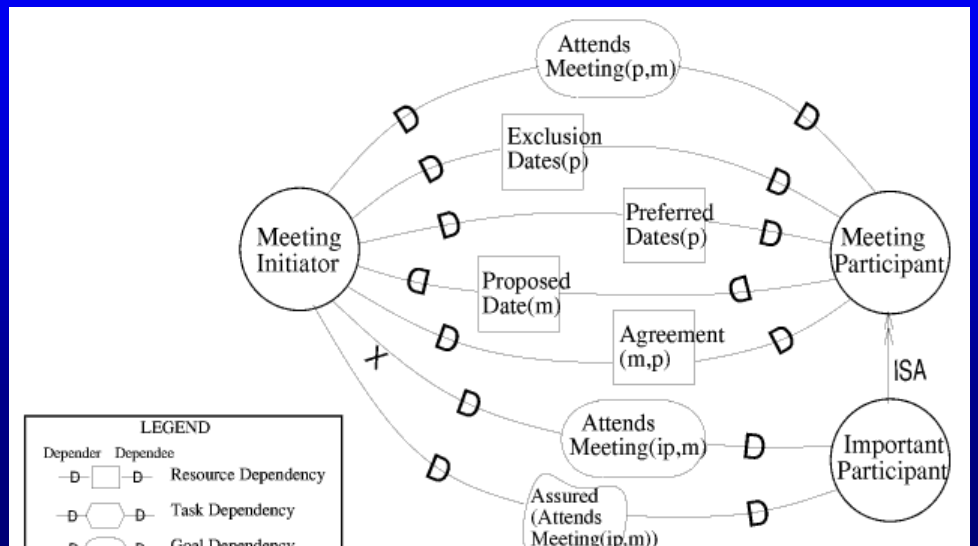


Meeting Scheduling Example

Agent Orientation as a Modelling Paradigm

2. Autonomy

- An agent has its own initiative, and can act independently. Consequently, for a modeller and from the viewpoint of other agents:
  - its behaviour is not fully predictable.
  - It is not fully knowable,
  - nor fully controllable.
- The behaviour of an agent can be partially characterized, despite autonomy, using intentional concepts.



Agent Orientation as a Modelling Paradigm

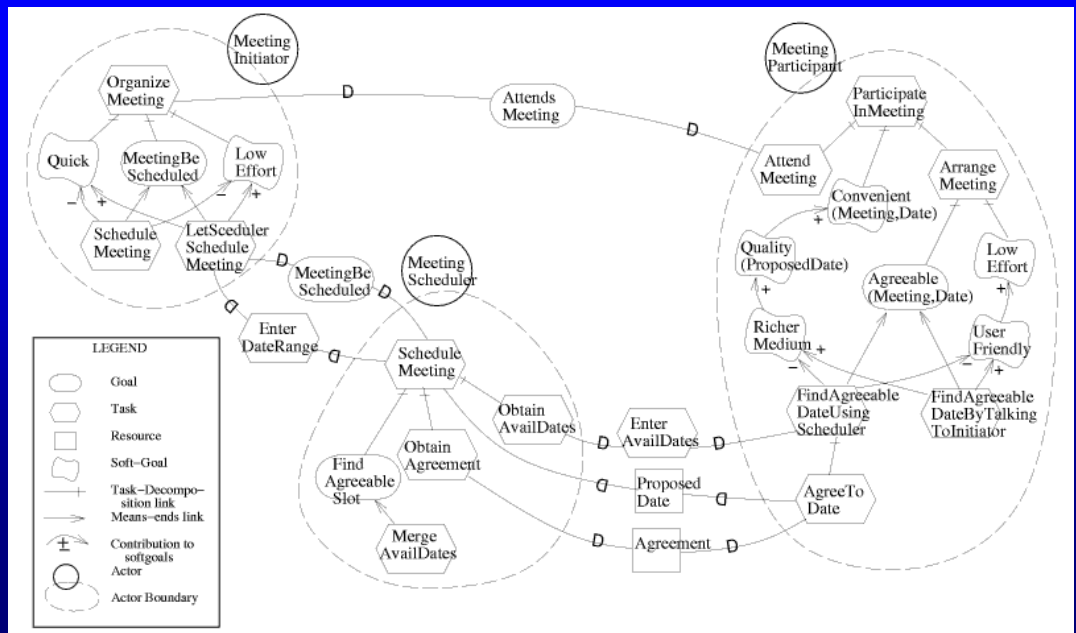
3. Sociality

- An agent is characterized by its relationships with other agents, and not by its intrinsic properties alone.
- Relationships among agents are complex and generally not reducible.
- Conflicts among many of the relationships that an agent participates in are not easily resolvable.
- Agents tend to have multi-lateral relationships, rather than one-way relationships.
- Agent relationships form an unbounded network
- Cooperation among agents cannot be taken for granted.
- Autonomy is tempered by sociality.

Agent Orientation as a Modelling Paradigm

4. Identity & Boundaries

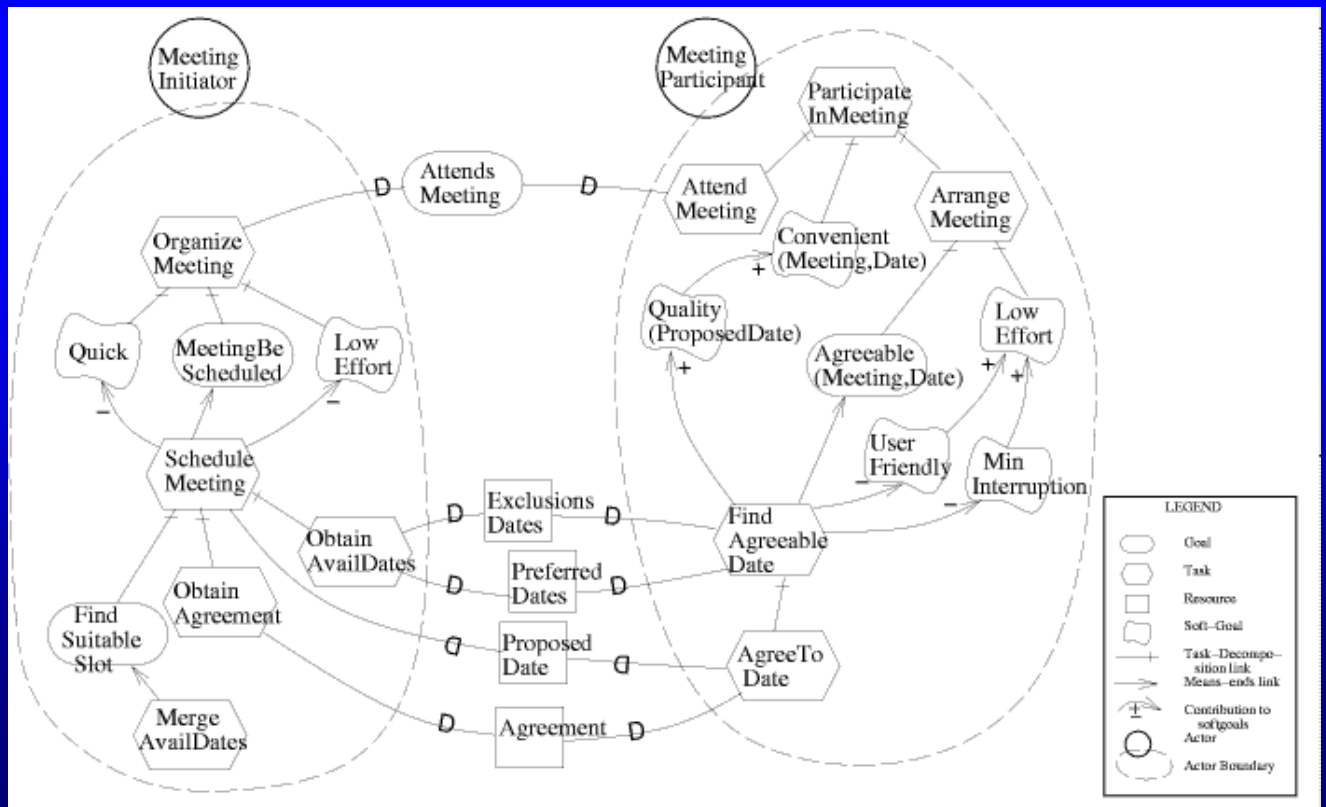
- Agents can be abstract, or physical.
- The boundaries, and thus the identity, of an agent are contingent and changeable.
- Agent, both physical and abstract, may be created and terminated.
- Agent behaviour may be classified, and generalized.



Agent Orientation as a Modelling Paradigm

5. Strategic Reflectivity

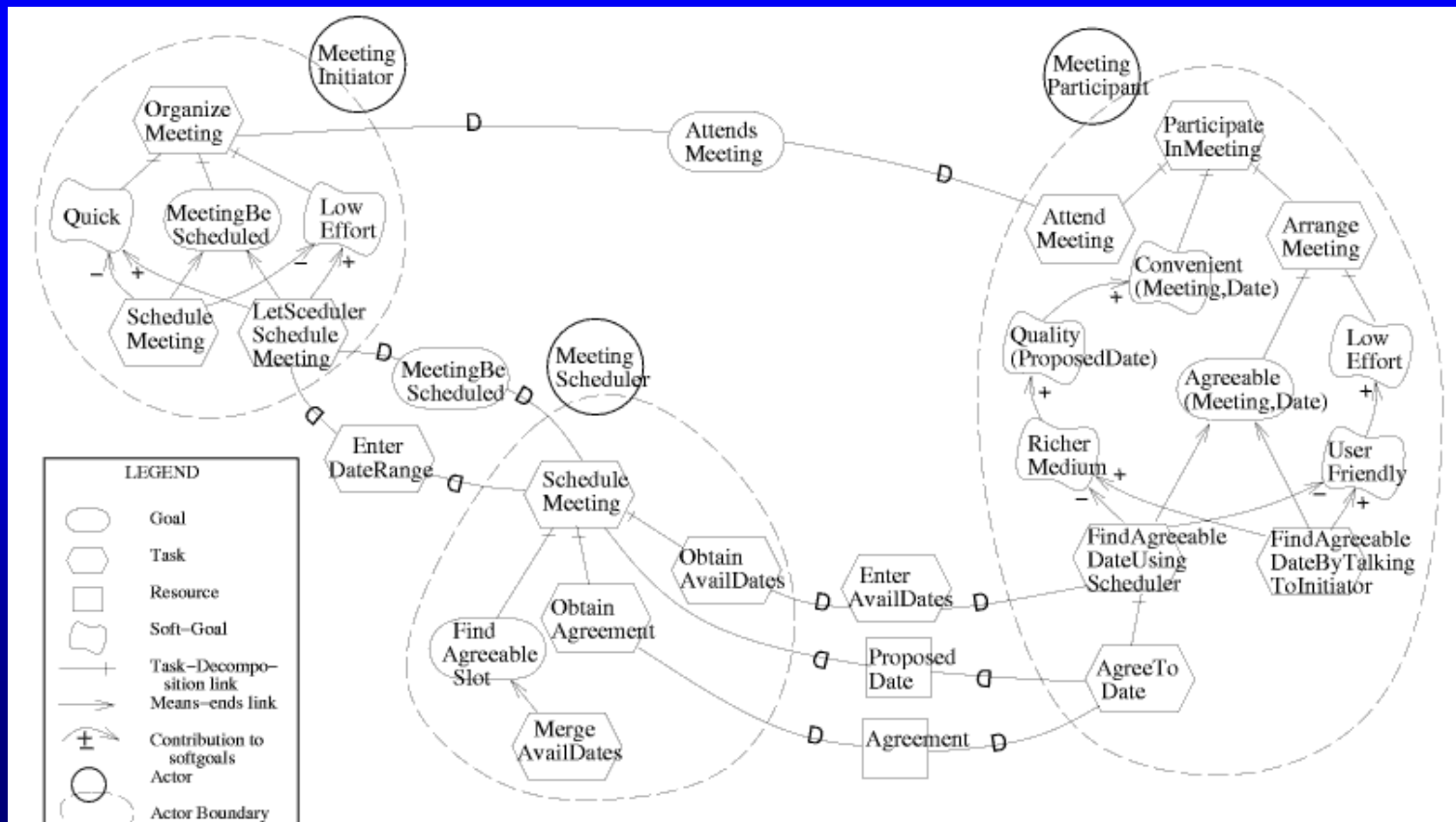
- Agents can reflect upon their own operations.
- Development world deliberations and decisions are usually strategic with respect to the operational world.
- The scope of reflectivity is contingent.



Agent Orientation as a Modelling Paradigm

6. Rational Self-Interest

- An agent strives to meet its goals.
- Self-interest is in a context of social relations.
- Rationality is bounded and partial.

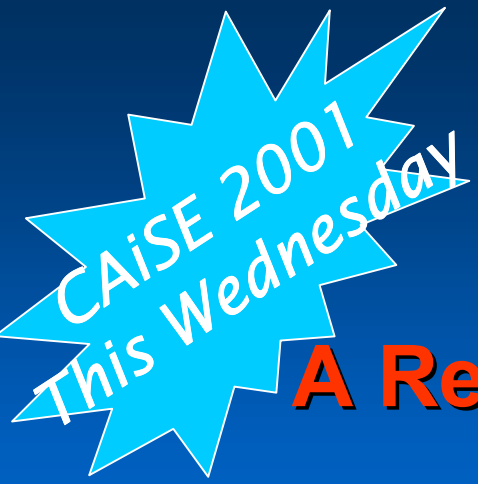


Points to remember

1. **AOIS (and IS) is not just about technology.**
   - Systems exist in a social/organizational environment.
   - ISD/SE continues to be intensely a human activity.
2. **Need to model relationships between machine and world.**
   - Requirements Engineering
3. **Use agent concepts for modeling, analysis, design, regardless of implementation technology.**
   - goals, means-ends, strat. deps., opportunities, vulnerabilities…

Now apply to …

Software Development throughout



A Requirement-Driven Development Methodology

Jaelson Castro † Manuel Kolp ‡ John Mylopoulos ‡

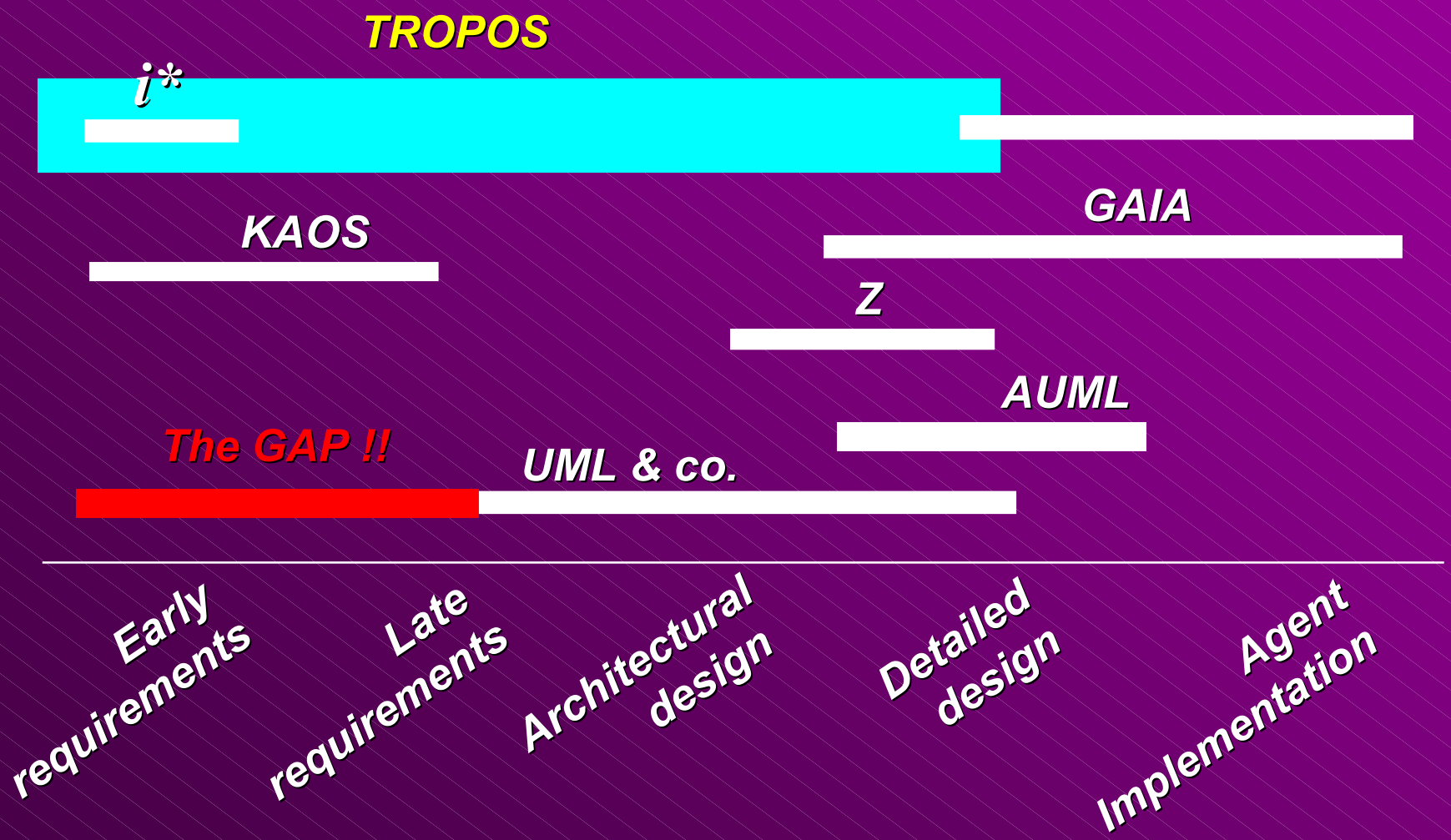
† Centro de Informática
Universidade Federal de Pernambuco
Recife 50732-970, Brazil

‡ Department of Computer Science
University of Toronto
Toronto M5S 3G4, Canada

CAiSE'01 - June 6 2001, Interlaken, Switzerland

Agent-Oriented Software Development

[J. Mylopoulos AOIS'99 Invited Talk]



Tropos & related projects

http://www.cs.toronto.edu/km/tropos

U. of Toronto, Canada

- John Mylopoulos
- Eric Yu
- Yves Lespérance
- Manuel Kolp
- Ariel Fuxman

RWTH Aachen, Germany

- Matthias Jarke
- Gerhard Lakemeyer
- Gunther Gans

U. of Trento/IRST, Italy

- Paolo Bresciani
- Paolo Giorgini
- Fausto Giunchiglia
- John Mylopoulos
- Anna Perini
- Marco Pistore
- Paolo Traverso

UFPE Recife, Brazil

- Jaelson Castro

Research Agenda

- **Ontology**
- **Formalization**
- **Analysis and reasoning**
- **Methodologies**
- **Knowledge Based Support**
  - Generic knowledge, e.g., common NFR goals, refinements, solution techniques (e.g., for security, safety,…)
  - Larger patterns
- **Tools**
- **Evaluation, Validation, Empirical studies**
- **Heterogeneous modelling frameworks**

Recent Work

- **Requirements -> architecturaql design** *STRAW01*
- **Trust & security** *Trust00*
- **GRL – part of URN proposal for ITU standard**
- **GRL+UCM** *STRAW01*
- **Intellectual property management** *submitted*
- **Others – Tropos** *AOIS@AA01* **i*+ConGolog** *Lesperance***+**

Points to remember

1. **AOIS (and IS) is not just about technology.**
   - Systems exist in a social/organizational environment.
   - ISD/SE continues to be intensely a human activity.
2. **Need to model relationships between machine and world.**
   - Requirements Engineering
3. **Use agent concepts for modeling, analysis, design, regardless of implementation technology.**
   - goals, means-ends, strat. deps., opportunities, vulnerabilities…
4. **AOIS means new ways for understanding problems, and for translating them into solutions.**
   - techniques for expressing, analyzing problems, solutions, and for matching them at each stage/level in development.