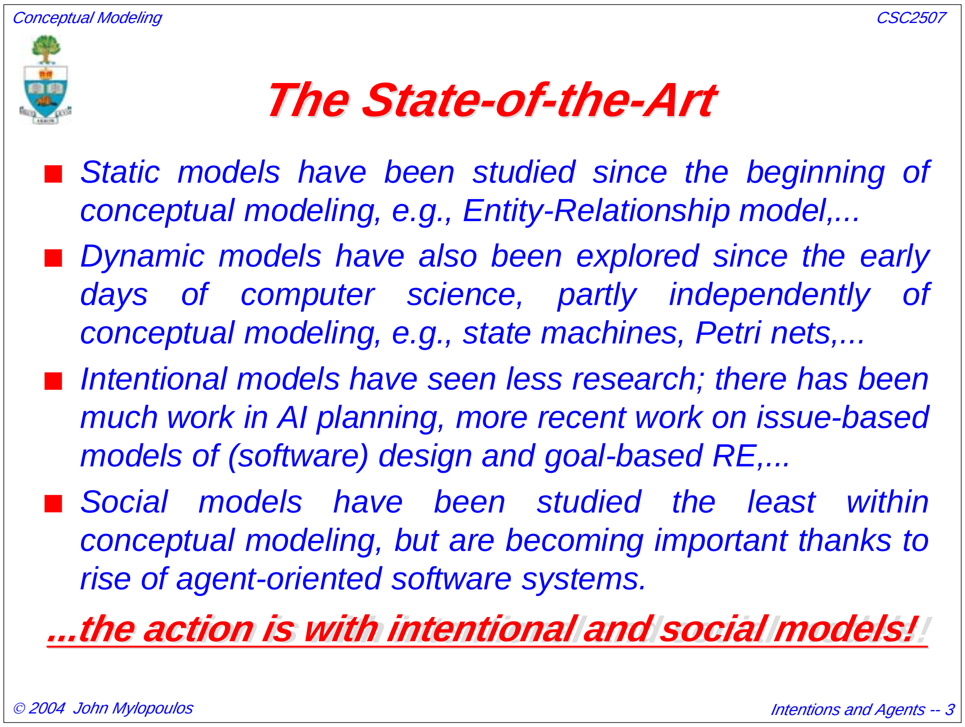
Intentions and Agents

From Entities and Relationships
to Goals and Agents
Intentions, Goals, Softgoals
Design Rationale
Agents and Social Settings
Speech Acts and Action Workflows
Actors and Social Dependencies

Types of Domains and Models Thereof

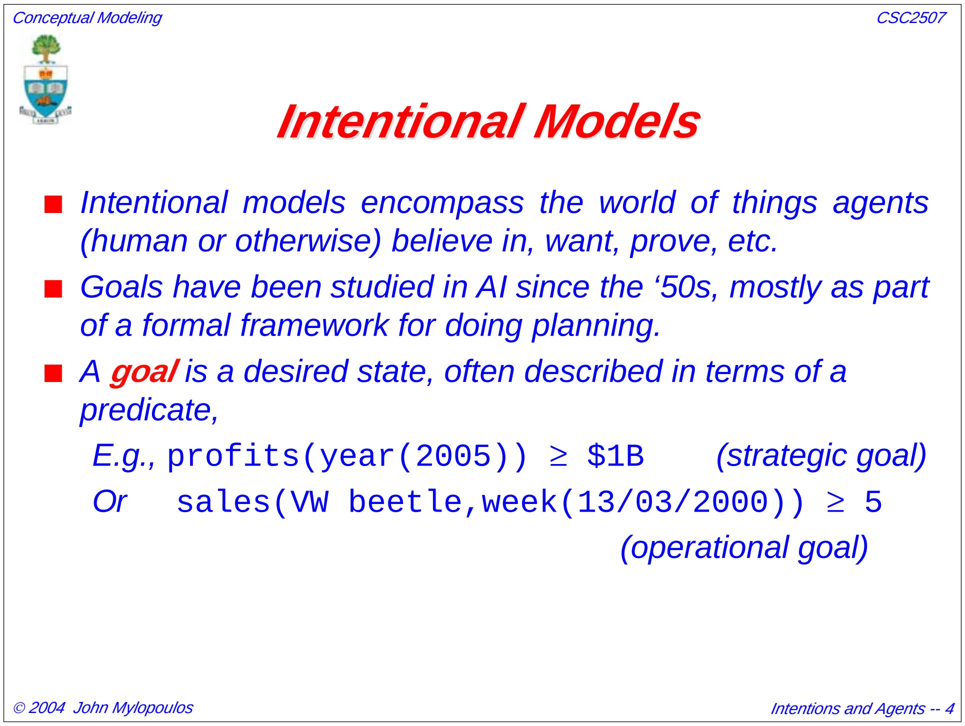
- ***Static models*** use concepts such as *Entity*, *Attribute*, *Relationship*, *Resource*,...
- ***Dynamic models*** described in terms of *Process*, *Activity*, *Action*, *Plan*, *Procedure*, *Event*,...or *State*, *Transition*,...
- ***Intentional models*** describe the world of things agents (human or otherwise) believe in, want, prove, argue about, e.g., *Issue*, *Goal*, *Softgoal*, *Supports*,…
- ***Social models*** describe social settings in terms of social relationships among agents, such as *Authority*, *Commitment*, *Responsibility*, *Actor*, *Position*, *Role*, *Goal/Task/Resource Dependency*,...
- ...Others...



The State-of-the-Art

- Static models have been studied since the beginning of conceptual modeling, e.g., Entity-Relationship model,...
- Dynamic models have also been explored since the early days of computer science, partly independently of conceptual modeling, e.g., state machines, Petri nets,...
- Intentional models have seen less research; there has been much work in AI planning, more recent work on issue-based models of (software) design and goal-based RE,...
- Social models have been studied the least within conceptual modeling, but are becoming important thanks to rise of agent-oriented software systems.

...the action is with intentional and social models!



Intentional Models

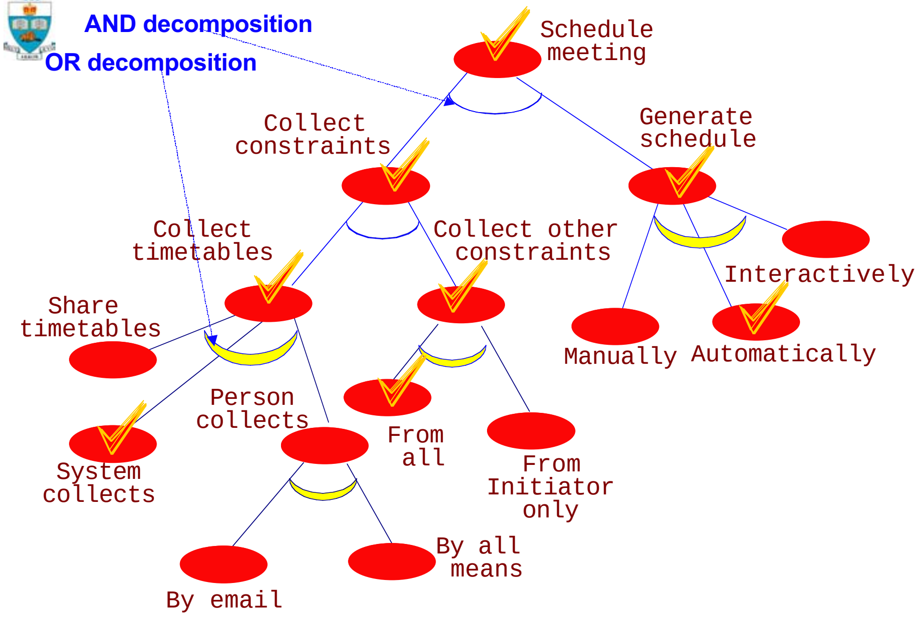
- Intentional models encompass the world of things agents (human or otherwise) believe in, want, prove, etc.
- Goals have been studied in AI since the ‘50s, mostly as part of a formal framework for doing planning.
- A ***goal*** is a desired state, often described in terms of a predicate,

  E.g., `profits(year(2005)) ≥ $1B` (strategic goal)

  Or `sales(VW beetle,week(13/03/2000)) ≥ 5` (operational goal)

AND/OR Goal Graphs

- Goals can be AND- or OR-decomposed to build AND/OR graphs.
- A simple procedure exists for AND/OR graphs for determining whether a root node of an AND/OR goal graph is solved/fulfilled, given that some other nodes of the graph have been found to be solved/fulfilled, or unsolvable/ unfulfillable.



AND decomposition

OR decomposition

Schedule meeting

Collect constraints

Generate schedule

Collect timetables

Collect other constraints

Interactively

Share timetables

Manually

Automatically

Person collects

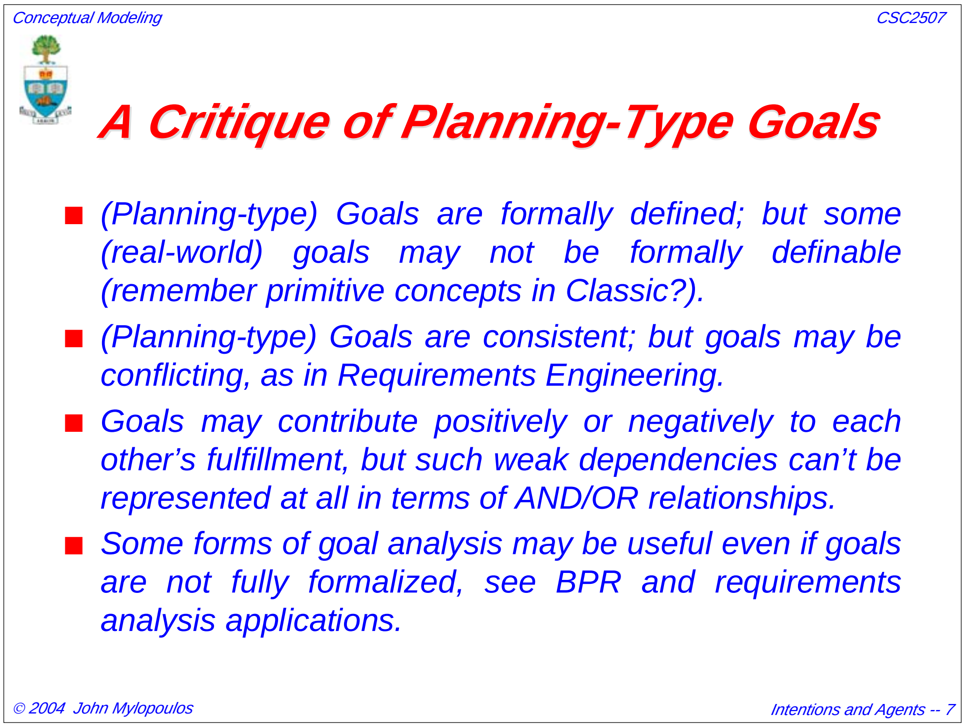
From all

System collects

From Initiator only

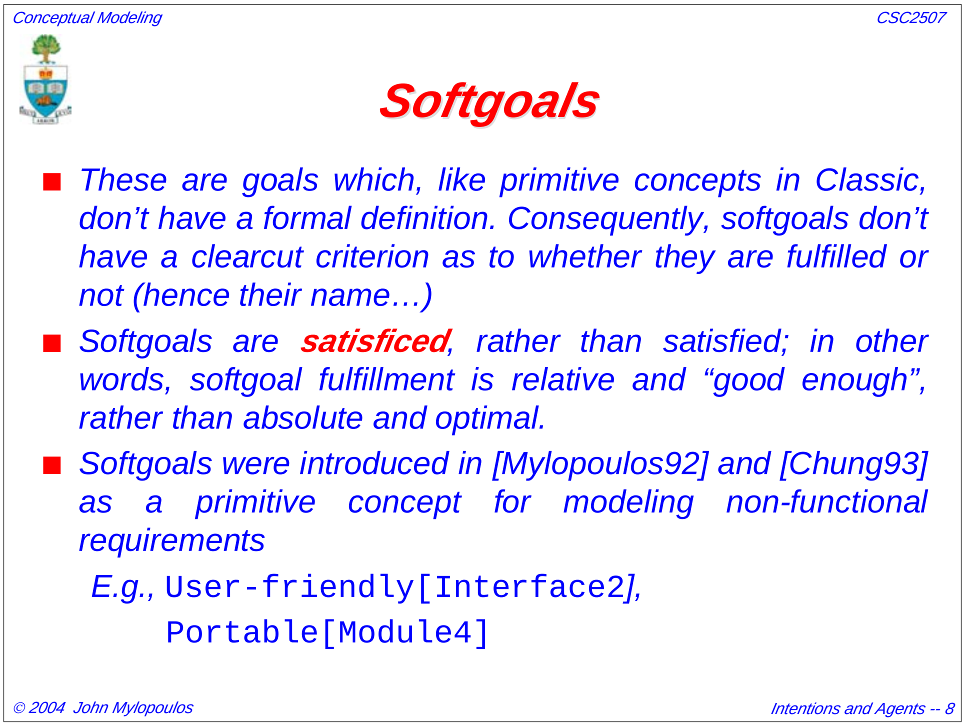
By all means

By email



A Critique of Planning-Type Goals

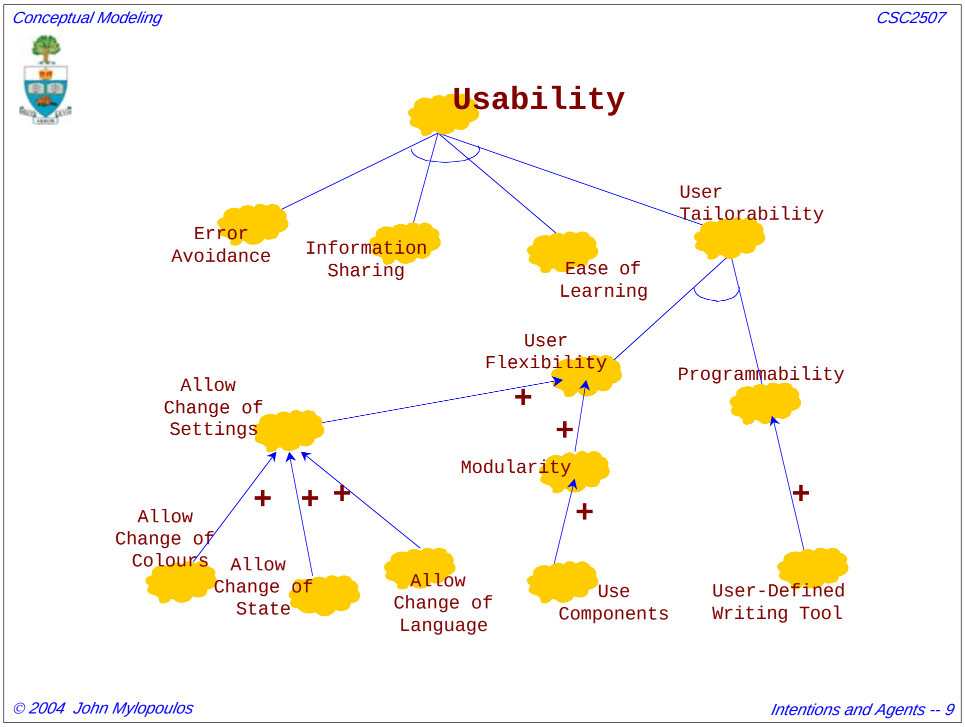
- *(Planning-type) Goals are formally defined; but some (real-world) goals may not be formally definable (remember primitive concepts in Classic?).*
- *(Planning-type) Goals are consistent; but goals may be conflicting, as in Requirements Engineering.*
- *Goals may contribute positively or negatively to each other's fulfillment, but such weak dependencies can't be represented at all in terms of AND/OR relationships.*
- *Some forms of goal analysis may be useful even if goals are not fully formalized, see BPR and requirements analysis applications.*



Softgoals

- *These are goals which, like primitive concepts in Classic, don't have a formal definition. Consequently, softgoals don't have a clearcut criterion as to whether they are fulfilled or not (hence their name…)*
- *Softgoals are* ***satisficed****, rather than satisfied; in other words, softgoal fulfillment is relative and "good enough", rather than absolute and optimal.*
- *Softgoals were introduced in [Mylopoulos92] and [Chung93] as a primitive concept for modeling non-functional requirements*

  E.g., `User-friendly[Interface2]`*,*
  `Portable[Module4]`

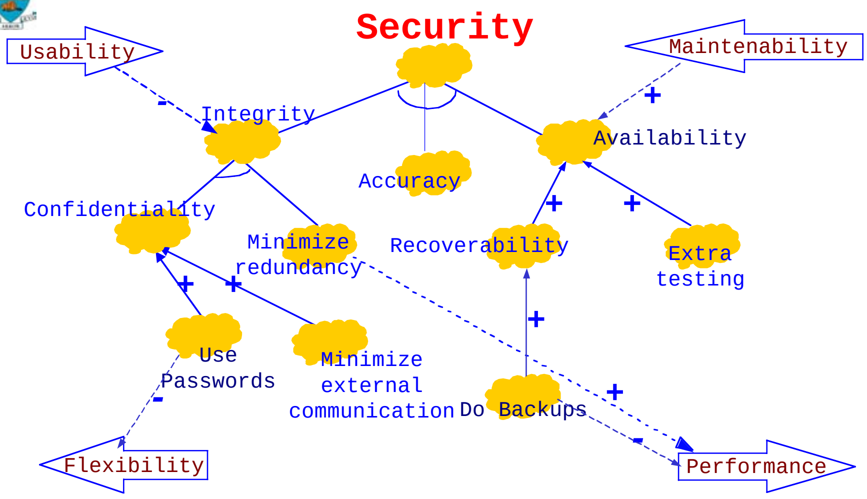


Goal Relationships

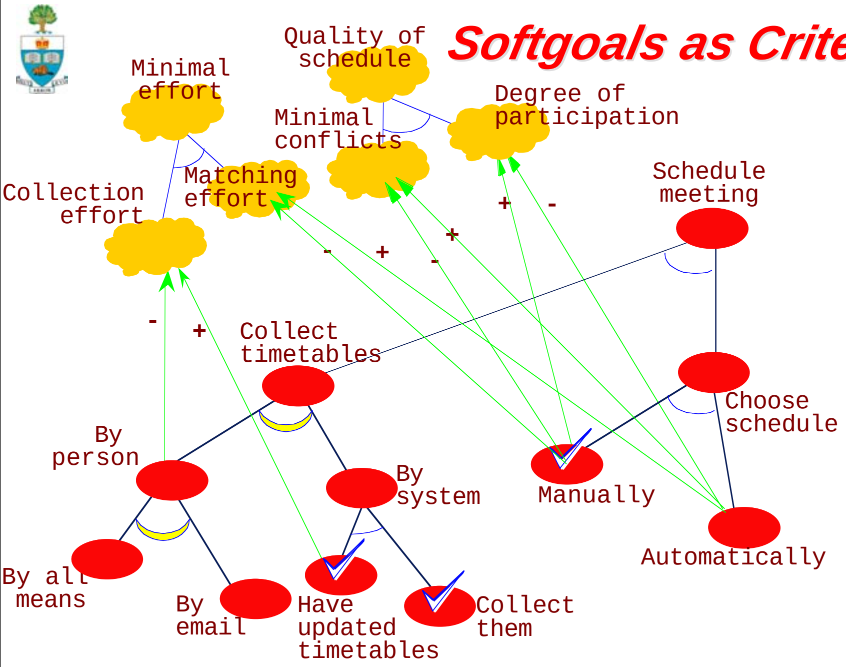
- *To arrive at a more qualitative framework for modeling goals, we also need to extend the set of relationships between goals beyond AND- and OR-relationships:*
  - ✓ *+ -- one goal contributes positively towards the fulfillment of another goal;*
  - ✓ *- -- one goal contributes negatively towards the fulfillment of another goal;*
  - ✓ *++ (--) -- one goal subsumes/negates another, I.e., if the first goal is fulfilled, the second is fulfilled/denied;*
- *With these enhancements, we can build goal models which could be useful for strategic business analysis or requirements analysis (as opposed to planning).*

Building Goal Dependency Graphs

- Start from one or more goals and/or softgoals $G_1, G_2, \dots, G_n$ which need to be fulfilled together.
- Analyze each, looking for ways to fulfill it through AND- or OR-decompositions, or through other refinements which contribute positively.
- Continue this process until there is enough positive support to fulfill all root nodes. At this point you have n disconnected goal trees $T(G_1), T(G_2), \dots, T(G_n)$.
- Identify positive and negative inter-tree influences, I.e., positive or negative relationships between goals g, g′ which belong to different goal trees.
- Repeat the analysis to see if root goals are fulfilled; if so, done, else continue the analysis.



Softgoals as Criteria



Issues,Positions and Arguments

- *Understanding decisions often involves asking **why?**:*
  - *"Why is this variable declared in this block?"*
  - *"Why did we decide to sell our Oshawa plant?" etc.*
- *To answer such questions, we need to represent, somehow, the **rationale** for particular decisions. This rationale may link particular decisions to operational goals, or operational goals to strategic ones, and strategic goals to their origins (stakeholders etc.)*
- *This is an important issue for Software Engineering in general, but is also important for strategic business analysis and other knowledge management applications.*

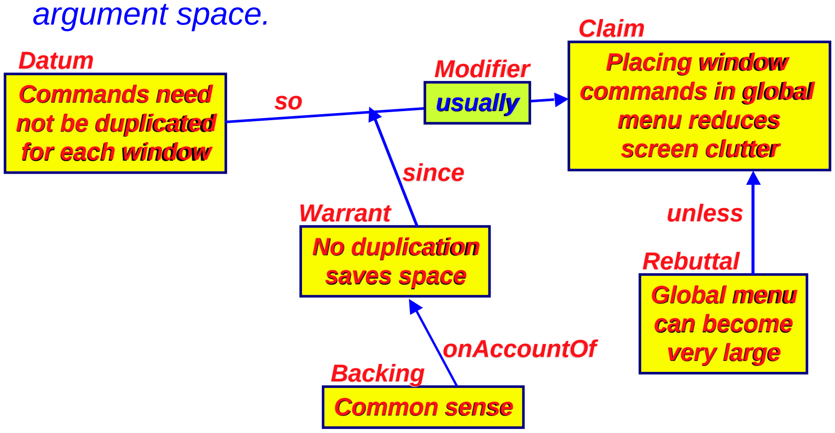
What do we Want to Represent?

These are progressively more refined models for capturing decision rationale:

- ***Arguments space*** -- a decision is associated with all relevant arguments, also their interrelations (e.g., `implies`, `supports`, `denies`, `qualifies`)
- ***Alternatives space*** -- alternatives and their arguments are made explicit; arguments about a particular alternative are differentiated from other arguments.
- ***Evaluation space*** -- evaluation measures are made explicit along with their arguments
- ***Criteria space*** -- now criteria used for evaluation are made explicit as well, along with their arguments
- ***Issues space*** -- issues are made explicit, along with their alternatives, evaluations, and criteria

Toulmin's Model of Argumentation

- Toulmin, a philosopher, proposed his model for describing arguments in 1958 [Toulmin58]. His model is limited to the argument space.



Datum

Commands need not be duplicated for each window

so

Modifier

usually

Claim

Placing window commands in global menu reduces screen clutter

since

Warrant

No duplication saves space

onAccountOf

Backing

Common sense

unless

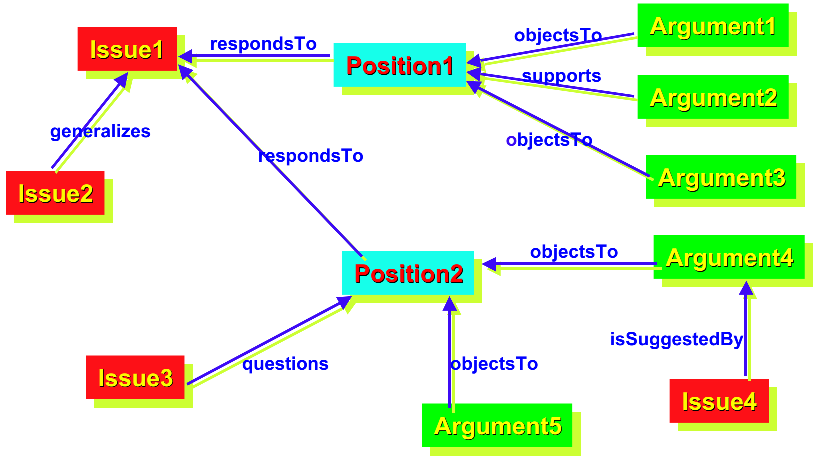
Rebuttal

Global menu can become very large

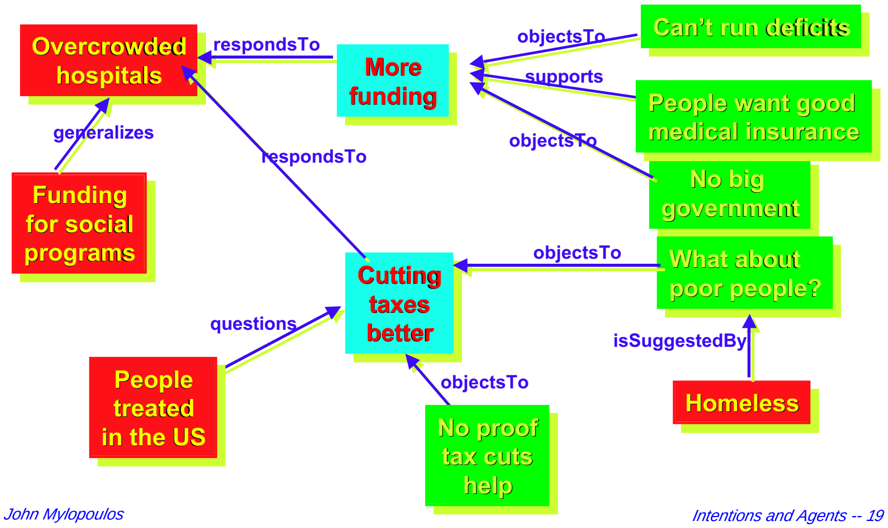
IBIS and gIBIS

- IBIS [Kunz70], the ***Issue-Based Information System***, is a first comprehensive attempt to capture decision rationale in Computer Science.
- gIBIS is the graphical version of IBIS, based on hypertext ideas [Conklin87].
- Basic idea is to capture the issues that were raised during the decision making process, the arguments pro and con particular positions and the resulting decision.
- Nodes of an IBIS network represent ***issues***, ***positions*** and ***arguments***.
- Framework can be used to expose faulty assumptions, unexplored alternatives and weak or missing rationale.

IBIS Representation of a Discussion



IBIS Representation of a Discussion



Overcrowded hospitals

respondsTo

More funding

objectsTo

Can't run deficits

supports

People want good medical insurance

objectsTo

No big government

generalizes

respondsTo

Funding for social programs

objectsTo

What about poor people?

Cutting taxes better

questions

isSuggestedBy

People treated in the US

objectsTo

Homeless

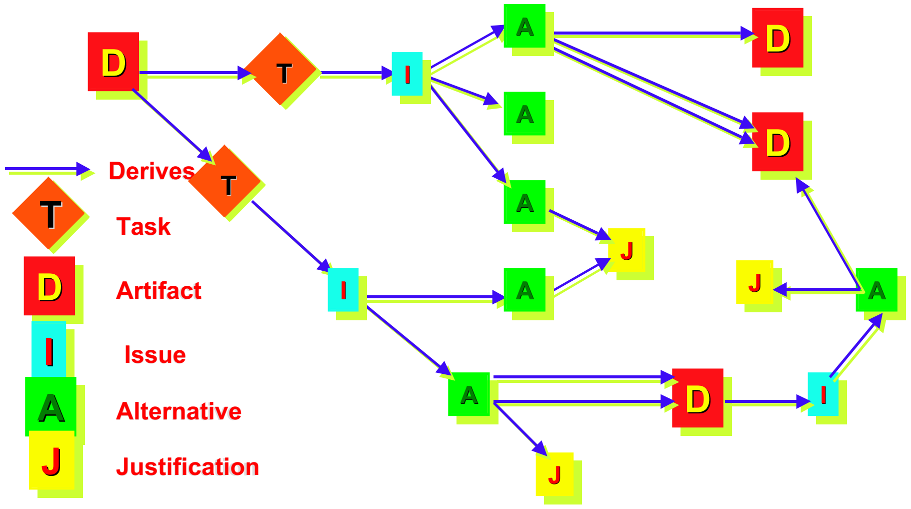
No proof tax cuts help

Potts and Bruns

- Proposal for capturing design rationale, IBIS-based.
- Design rationale represented as a network of ***design artifact*** and ***deliberation*** concepts.
- Design artifacts document (elements of) the design.
- Deliberations are ***issues***, ***alternatives*** or ***justifications.***
- ***Planetext*** is the graphical implementation of the system, using hypertext links.
- A Planetext network can actually be translated into Horn clauses and fed into a Prolog engine which analyzes it looking for inconsistencies.

[Potts88]

A Potts and Bruns Network



Derives

T Task

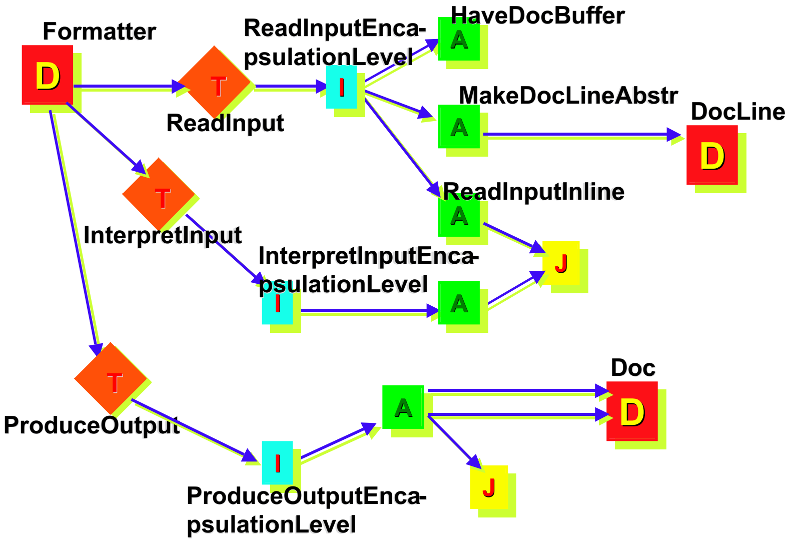
D Artifact

I Issue

A Alternative

J Justification

Potts and Bruns: An Example



Formatter

ReadInput

ReadInputEncapsulationLevel

HaveDocBuffer

MakeDocLineAbstr

DocLine

ReadInputInline

InterpretInput

InterpretInputEncapsulationLevel

Doc

ProduceOutput

ProduceOutputEncapsulationLevel

Questions, Options, Criteria (QOC)

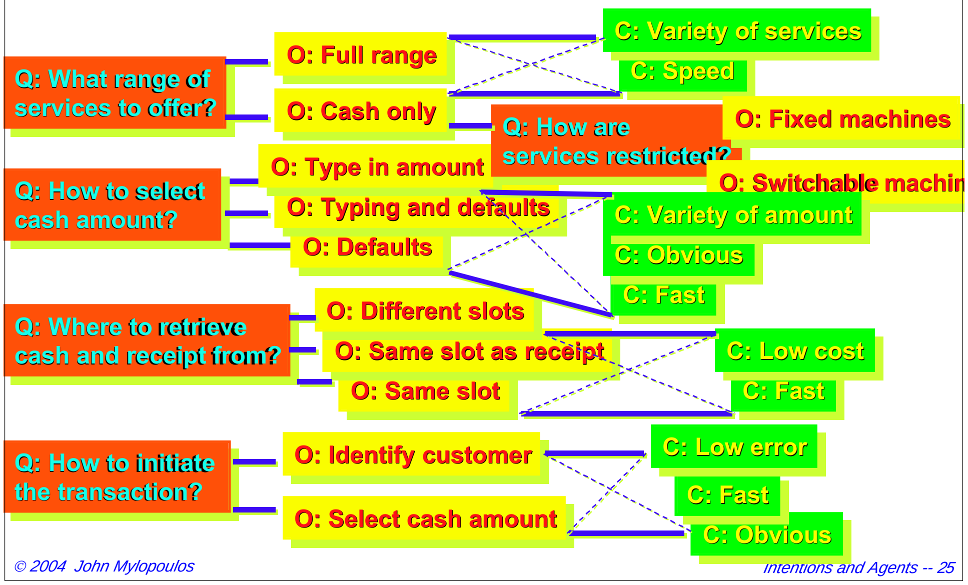
- Offers a framework for exploring design spaces, based on the comparison of alternative options.
- Extends Potts&Bruns framework with **goals**, **criteria**.
- A QOC network is composed of:
  - ✓ **Questions** -- pose key issues;
  - ✓ **Options** -- possible alternatives;
  - ✓ **Criteria** -- for selecting among options;
  - ✓ **Arguments** -- support/challenge a criterion.
- DRL is an implementation of QOC. QOC does not “capture design”; rather, it is supposed **to be part of** the design process

[McLean91]

Criteria

- A **criterion**
  - ✓ Measures a property that the designer **controls indirectly** by choosing an option;
  - ✓ Must be **unconditional**, i.e., the better the criterion, the better the design;
  - ✓ Must be **evaluative** in that it measures a property of the artifact;
- Criteria must be justified by referencing other criteria.
- **Bridging criteria** relate design decisions to general criteria, e.g., throughput vs performance.
- **Note:** Potts&Bruns issues are the result of a design decision; QOC options are the result of a question (issue)

QOC Diagram for ATM Design



Q: What range of services to offer?

O: Full range

O: Cash only

C: Variety of services

C: Speed

Q: How are services restricted?

O: Fixed machines

O: Switchable machin

Q: How to select cash amount?

O: Type in amount

O: Typing and defaults

O: Defaults

C: Variety of amount

C: Obvious

C: Fast

Q: Where to retrieve cash and receipt from?

O: Different slots

O: Same slot as receipt

O: Same slot

C: Low cost

C: Fast

Q: How to initiate the transaction?

O: Identify customer

O: Select cash amount

C: Low error

C: Fast

C: Obvious

Social Ontologies

- *These are collections of concepts for modeling social settings, e.g., a family, a university, or a research group.*
- *The units of a social setting are **actors**, such as **agents**, **positions** and **roles**. Actors may be atomic, such as Maria (...a person), InfoSleuth (…a program), or composite, such as the recruiting committee, DCS, UofT.*
- *Social relationships are inherently **intentional**. This means that they constrain the actors they relate over many possible worlds, rather than just the here-and-now world.*
- *For example, OnTopOf(projector,table) is extensional. But Mother(maria,george), Promise(attila,greg,a) are intentional in that they constrain the behaviour of their actors in future states of the world. The same holds for relationships such as owns, reportsTo,...*

Modeling Social Settings

Two complementary ways to model social settings:

- ***Work Action Approach*** *-- focuses on* ***speech acts*** *as means for changing the intentional state of actors; for example, a* `request` *speech act, if successful, gets the actor being requested to commit to fulfill the request.*
- ***Social Dependency Approach*** *-- represent explicitly actor* ***goals*** *and* ***inter-actor dependencies*** *and how these constrain social activity. For example, a manager depends on her engineers to deliver designs on time, while the engineers depend on the manager to give them a raise for their good work.*

The Action Workflow Framework

- *Some communication actions are intended to change the state of the social setting, such as*
  - ✓ *Asking someone for information;*
  - ✓ *Asking someone to have something done;*
  - ✓ *Having someone commit to do something etc.*
- *Such* ***social*** *actions are called* ***speech acts*** *[Searle69].*
- *Speech acts were originally proposed as a new approach to Linguistics, claiming that some utterances can best be understood as actions towards achieving some goal*

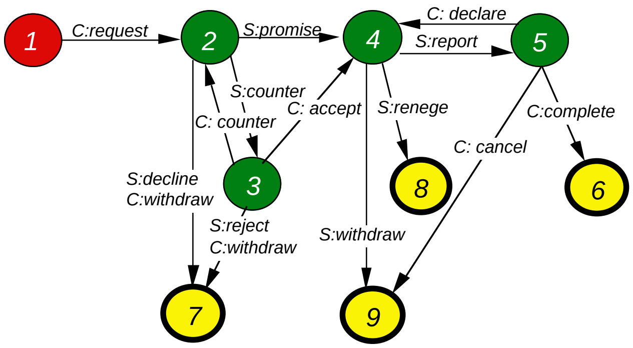
  e.g., "It's cold in here" means something like "someone, turn up the thermostat, please" (…at least when a king says it in the presence of his court.)

Speech Acts and Conversations

- [Winograd86] introduced speech act ideas to system design by arguing that in designing a human-computer system we need to focus on the interaction, rather than the human or the computer.
- Offered a substantial critique of AI and other areas of computer science for focusing too much on one or the other side, rather than the interaction itself.
- Proposed to use speech acts to model such interactions.
- This work was fundamental in establishing **Computer-Supported Collaborative Work** (CSCW) as an autonomous research area within Computer Science, also produced one of the first workflow products (**the Coordinator**).

The Basic Molecule of Social Action

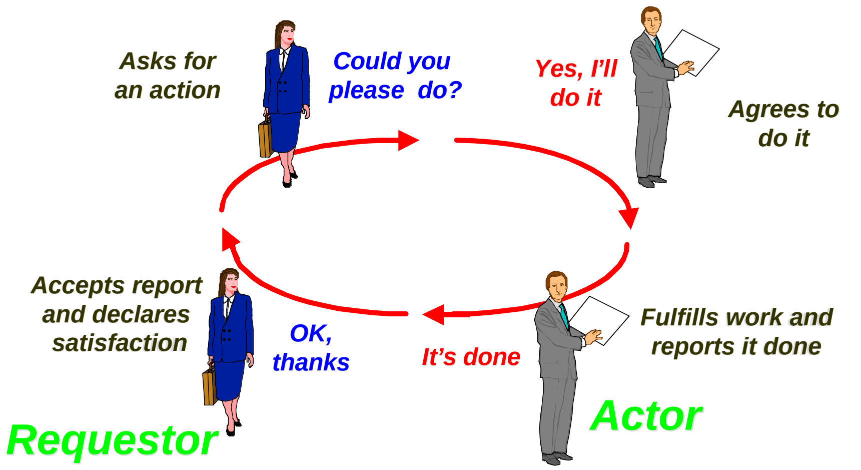
Think of this as the basic social action “molecule”, out of which one can build social processes



C: client
S: server

The Action Workflow Framework

A simpler action molecule:



Asks for an action

Could you please do?

Yes, I'll do it

Agrees to do it

Fulfills work and reports it done

It's done

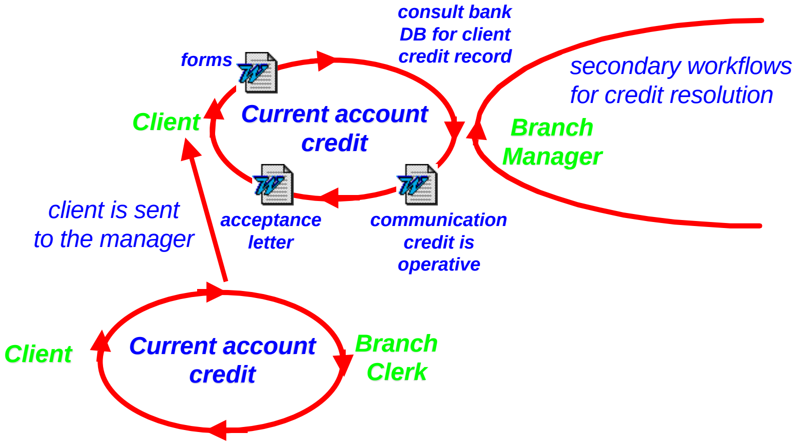
OK, thanks

Accepts report and declares satisfaction

Requestor

Actor

Getting a New Credit Account



forms

consult bank DB for client credit record

secondary workflows for credit resolution

Client

Current account credit

Branch Manager

acceptance letter

communication credit is operative

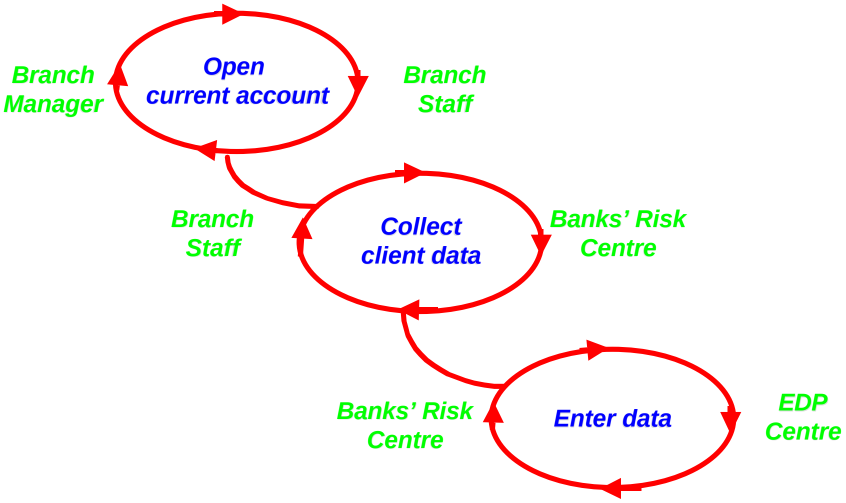
client is sent to the manager

Client

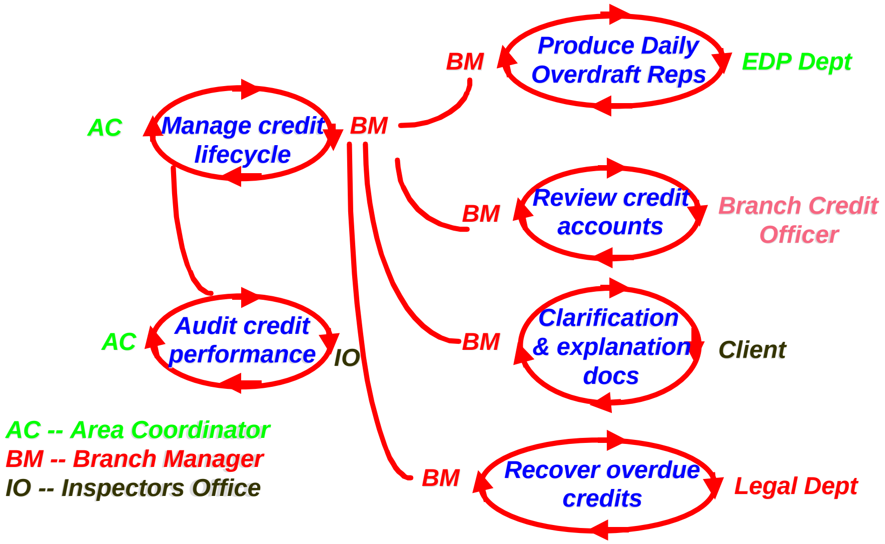
Current account credit

Branch Clerk

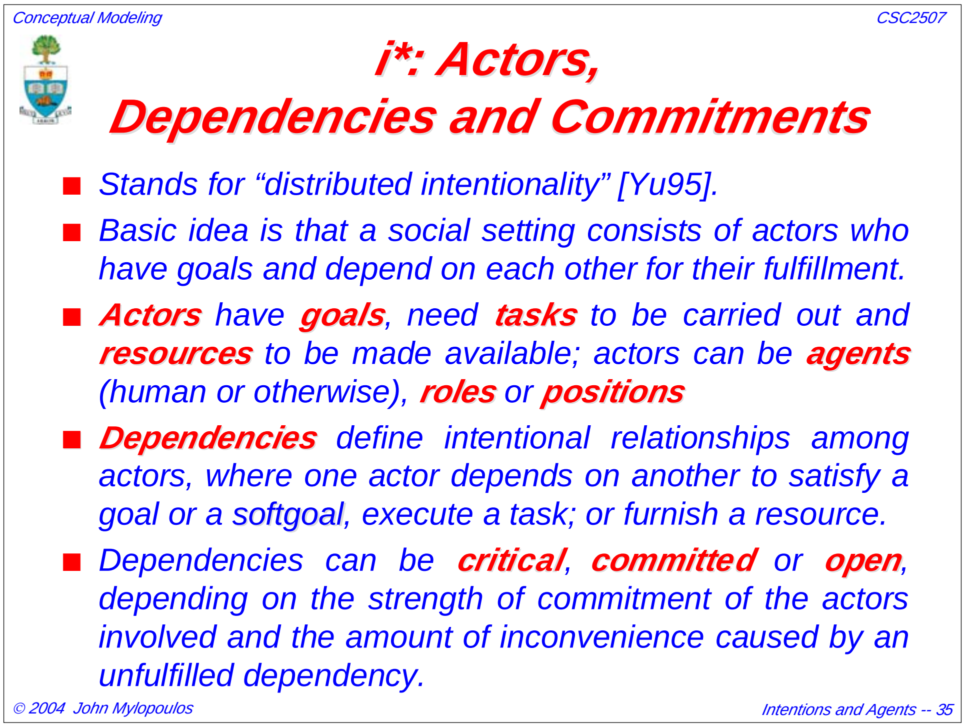
Opening a New Credit Account



Complete Credit Account Workflow

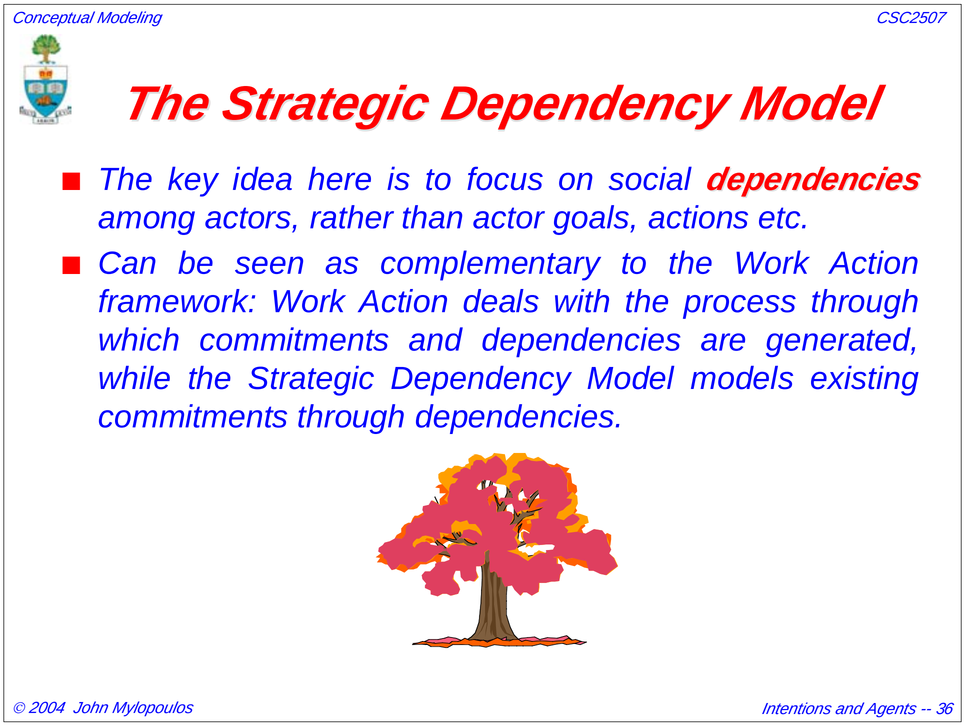


AC -- Area Coordinator
BM -- Branch Manager
IO -- Inspectors Office



i*: Actors, Dependencies and Commitments

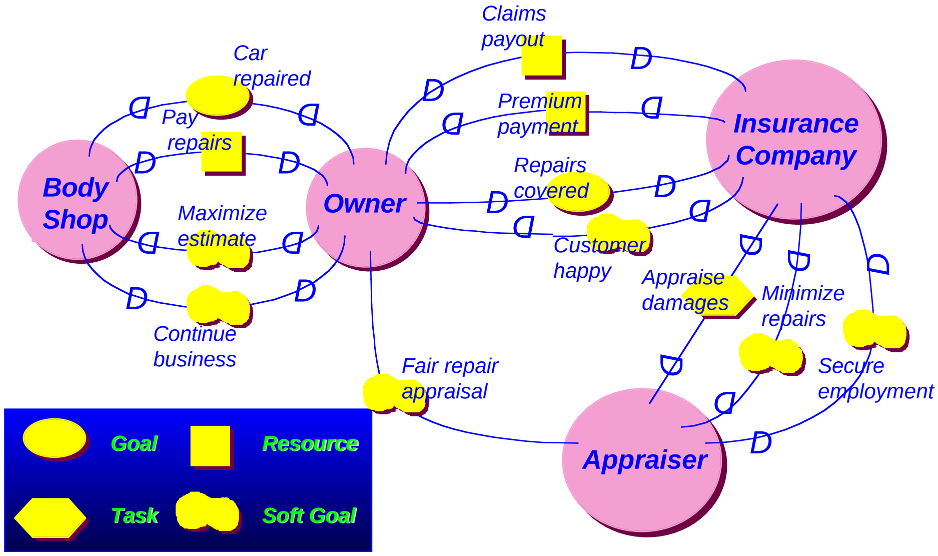
- Stands for “distributed intentionality” [Yu95].
- Basic idea is that a social setting consists of actors who have goals and depend on each other for their fulfillment.
- ***Actors*** have ***goals***, need ***tasks*** to be carried out and ***resources*** to be made available; actors can be ***agents*** (human or otherwise), ***roles*** or ***positions***
- ***Dependencies*** define intentional relationships among actors, where one actor depends on another to satisfy a goal or a softgoal, execute a task; or furnish a resource.
- Dependencies can be ***critical***, ***committed*** or ***open***, depending on the strength of commitment of the actors involved and the amount of inconvenience caused by an unfulfilled dependency.



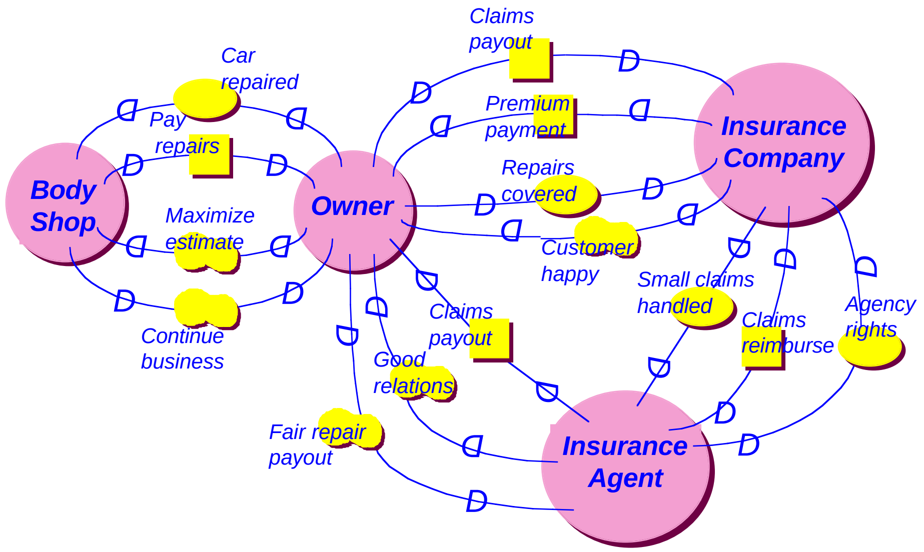
The Strategic Dependency Model

- The key idea here is to focus on social ***dependencies*** among actors, rather than actor goals, actions etc.
- Can be seen as complementary to the Work Action framework: Work Action deals with the process through which commitments and dependencies are generated, while the Strategic Dependency Model models existing commitments through dependencies.

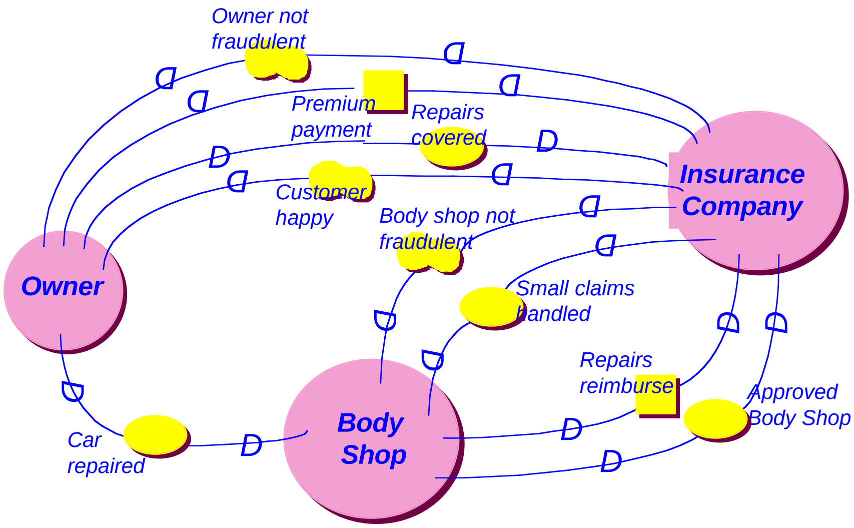
The Strategic Dependency Model



"...Let the Agent Handle it..."



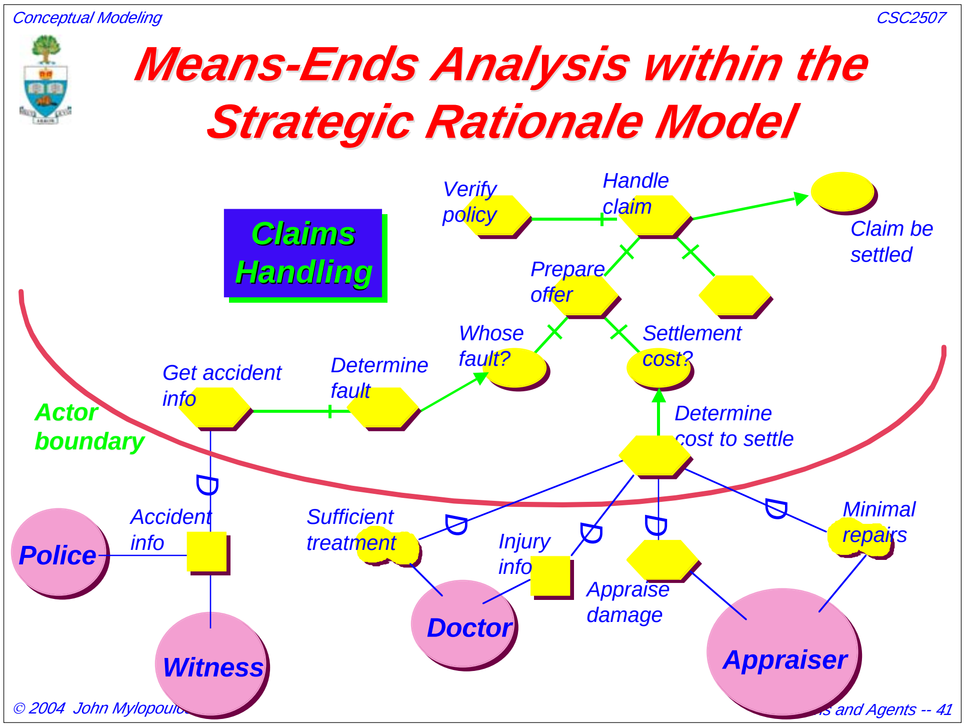
"... Let the Body Shop Handle it..."



The Need to Capture Rationale

How do actors' goals lead to social dependencies?

- **Means-ends** analysis can be used to relate goals to tasks that can satisfy these goals: "Given goal (end) G, how can I decompose it (means) in order to find a way to fulfill it".
- Means-ends analysis has been used in AI since GPS (General Problem Solver) proposed by Simon, Newell and Shaw in the '50s.
- **Task decomposition** links relate tasks to other, component tasks
- Tasks can also be decomposed into goals.



Means-Ends Analysis within the Strategic Rationale Model

Claims Handling

Verify policy

Handle claim

Claim be settled

Prepare offer

Whose fault?

Settlement cost?

Get accident info

Determine fault

Determine cost to settle

Actor boundary

Police

Accident info

Witness

Sufficient treatment

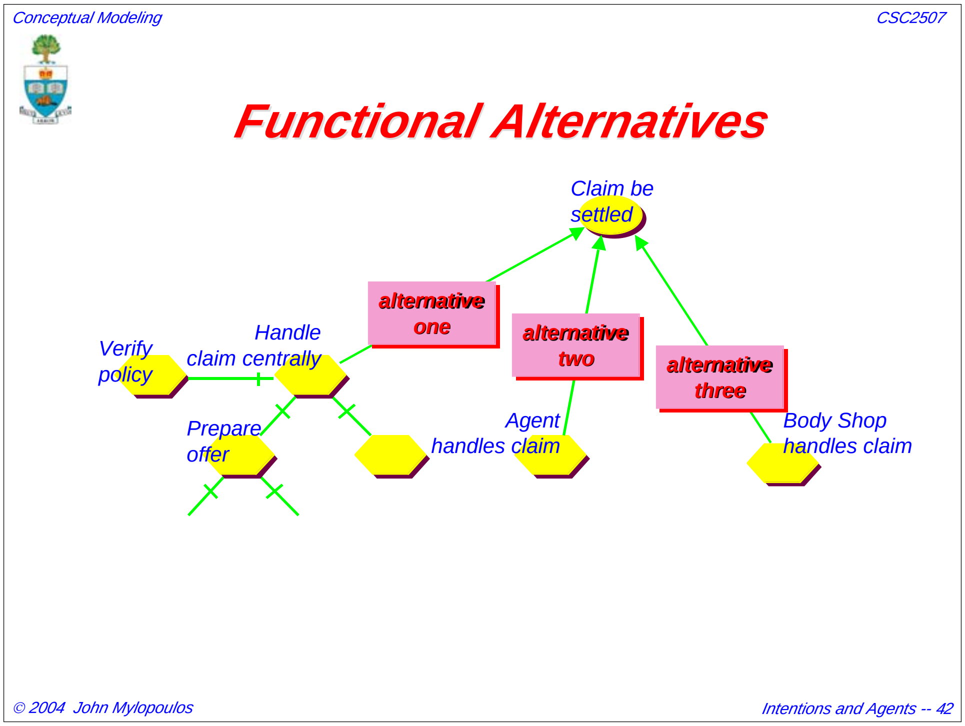
Injury info

Doctor

Appraise damage

Minimal repairs

Appraiser



Functional Alternatives

Claim be settled

alternative one

alternative two

alternative three

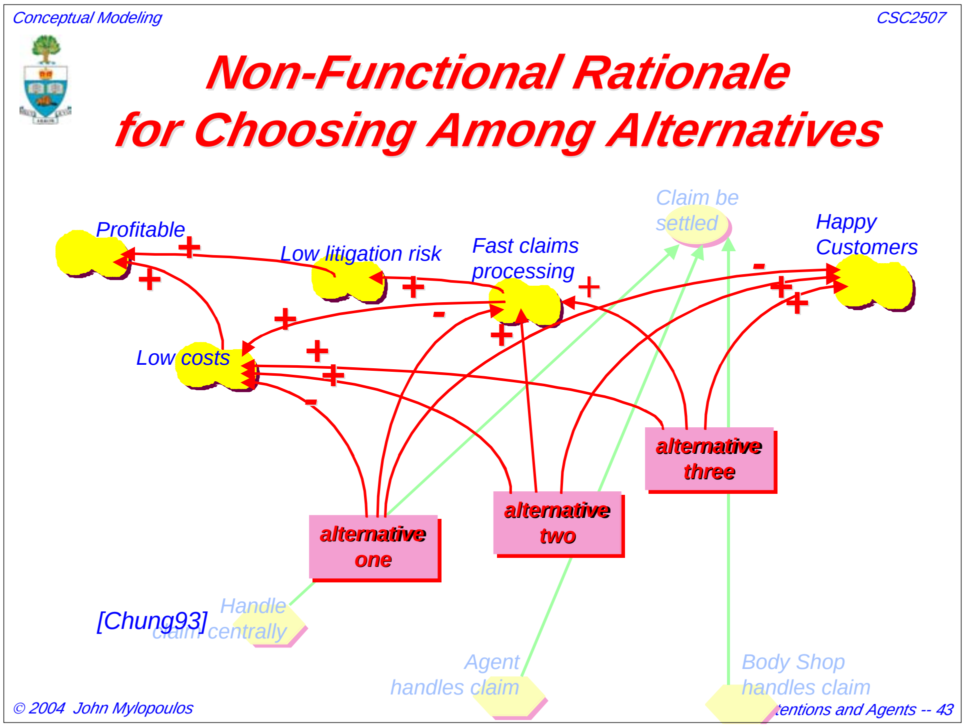
Verify policy

Handle claim centrally

Prepare offer

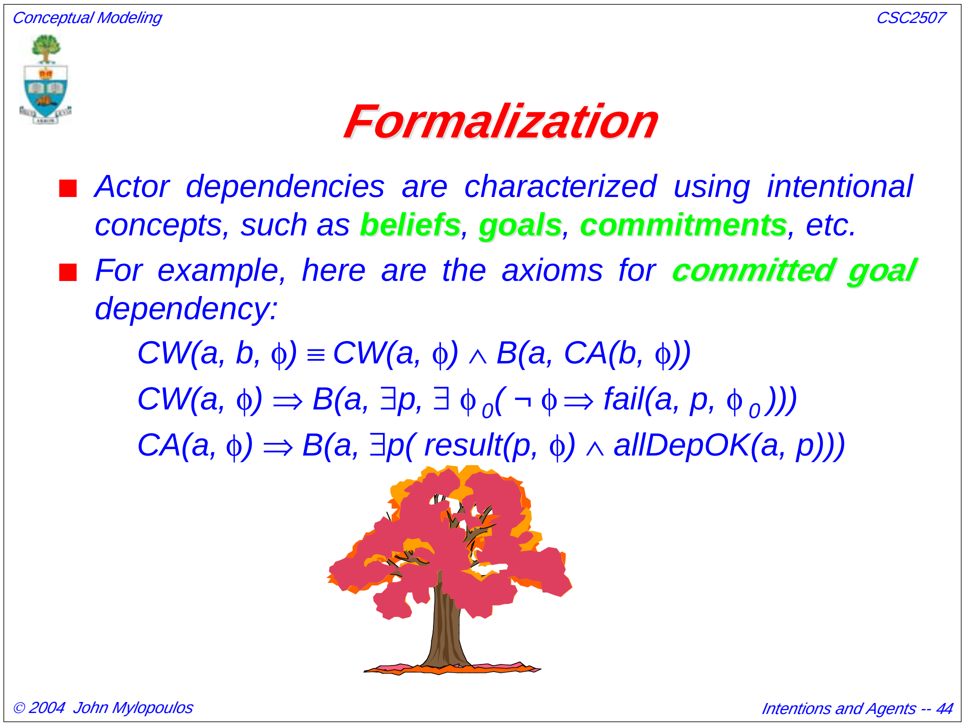
Agent handles claim

Body Shop handles claim



Non-Functional Rationale for Choosing Among Alternatives

[Chung93]



Formalization

- Actor dependencies are characterized using intentional concepts, such as **beliefs**, **goals**, **commitments**, etc.
- For example, here are the axioms for ***committed goal*** dependency:

$$CW(a, b, \phi) \equiv CW(a, \phi) \wedge B(a, CA(b, \phi))$$

$$CW(a, \phi) \Rightarrow B(a, \exists p, \exists \phi_0(\neg \phi \Rightarrow fail(a, p, \phi_0)))$$

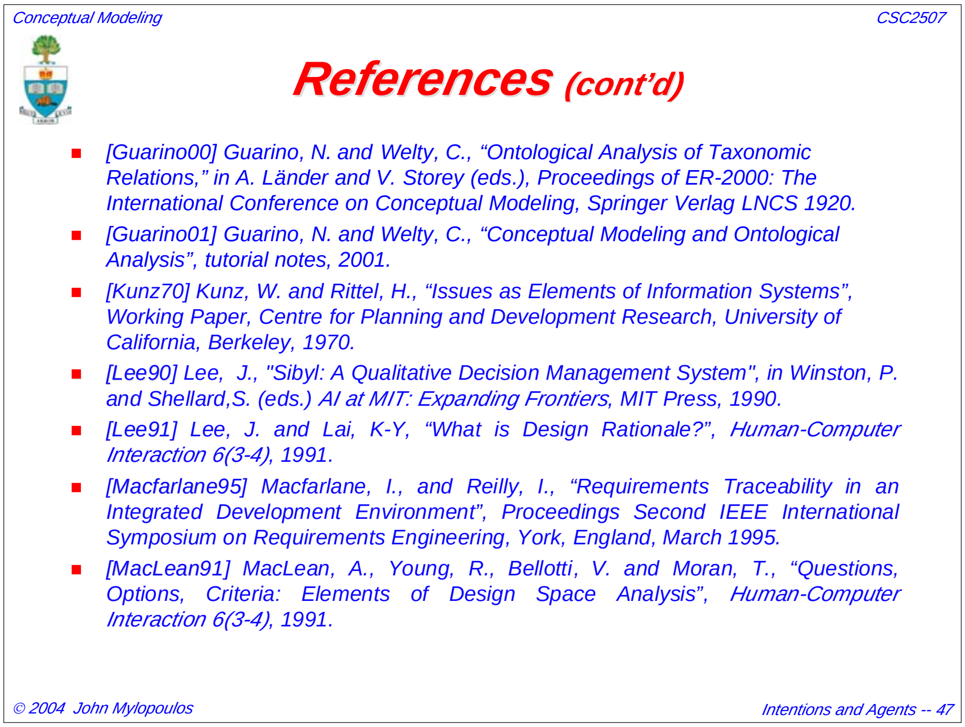
$$CA(a, \phi) \Rightarrow B(a, \exists p(result(p, \phi) \wedge allDepOK(a, p)))$$

Tropos: A Formal Extension of *i**

- Although the primitive concepts of *i** are defined formally, *i** is not an expressive language because it doesn't support the specification of constraints, invariants, pre/post-conditions for different elements of an *i** model.
- Tropos extends the expressiveness of *i** , adopting ideas from KAOS.
- Most notably, the Tropos specification language includes a temporal logic inspired by KAOS.
- Actors, goals, actions, entities, relationships are described both statically and dynamically.
- …stay tuned…

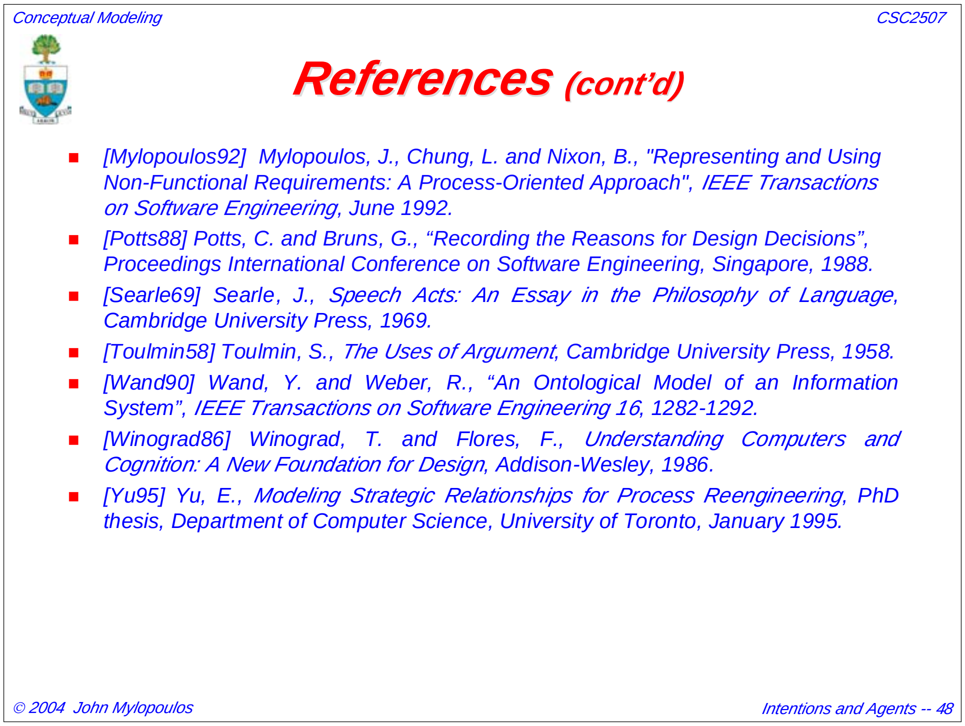
References

- [Begeman88] Begeman, M., Conklin, E., "The Right Tool for the Right Job", *Byte*, October 1988.
- [Bunge77] Bunge,M., *Treatise on Basic Philosophy: Ontology I, The Furniture of the World*, Reidel, 1977.
- [Chung93] Chung, L., *Representing and Using Non-Functional Requirements: A Process-Oriented Approach*, PhD thesis, Department of Computer Science, University of Toronto, June 1993.
- [Conklin87] Conklin, E., "Hypertext: An Introduction," *IEEE Computer*, September 1987.
- [Conklin91] Conklin, E., and Burgess-Yakemovic, K., "A Process-Oriented Approach to Design Rationale", *Human-Computer Interaction 6(3-4)*, 1991.
- [Gotel95] Gotel, O., Finkelstein, A., "Contribution Structures", Proceedings Second IEEE International Symposium on Requirements Engineering, York, England, March 1995.
- [Gruber91] Gruber, T., "The Role of a Common Ontology in Achieving Sharable, Reusable Knowledge Bases," Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning, Cambridge, 601-602.



References (cont'd)

- [Guarino00] Guarino, N. and Welty, C., “Ontological Analysis of Taxonomic Relations,” in A. Länder and V. Storey (eds.), Proceedings of ER-2000: The International Conference on Conceptual Modeling, Springer Verlag LNCS 1920.
- [Guarino01] Guarino, N. and Welty, C., “Conceptual Modeling and Ontological Analysis”, tutorial notes, 2001.
- [Kunz70] Kunz, W. and Rittel, H., “Issues as Elements of Information Systems”, Working Paper, Centre for Planning and Development Research, University of California, Berkeley, 1970.
- [Lee90] Lee, J., "Sibyl: A Qualitative Decision Management System", in Winston, P. and Shellard,S. (eds.) AI at MIT: Expanding Frontiers, MIT Press, 1990.
- [Lee91] Lee, J. and Lai, K-Y, “What is Design Rationale?”, Human-Computer Interaction 6(3-4), 1991.
- [Macfarlane95] Macfarlane, I., and Reilly, I., “Requirements Traceability in an Integrated Development Environment”, Proceedings Second IEEE International Symposium on Requirements Engineering, York, England, March 1995.
- [MacLean91] MacLean, A., Young, R., Bellotti, V. and Moran, T., “Questions, Options, Criteria: Elements of Design Space Analysis”, Human-Computer Interaction 6(3-4), 1991.



References (cont'd)

- [Mylopoulos92] Mylopoulos, J., Chung, L. and Nixon, B., "Representing and Using Non-Functional Requirements: A Process-Oriented Approach", IEEE Transactions on Software Engineering, June 1992.
- [Potts88] Potts, C. and Bruns, G., “Recording the Reasons for Design Decisions”, Proceedings International Conference on Software Engineering, Singapore, 1988.
- [Searle69] Searle, J., Speech Acts: An Essay in the Philosophy of Language, Cambridge University Press, 1969.
- [Toulmin58] Toulmin, S., The Uses of Argument, Cambridge University Press, 1958.
- [Wand90] Wand, Y. and Weber, R., “An Ontological Model of an Information System”, IEEE Transactions on Software Engineering 16, 1282-1292.
- [Winograd86] Winograd, T. and Flores, F., Understanding Computers and Cognition: A New Foundation for Design, Addison-Wesley, 1986.
- [Yu95] Yu, E., Modeling Strategic Relationships for Process Reengineering, PhD thesis, Department of Computer Science, University of Toronto, January 1995.