# *From Entities and Relationships to Social Actors and Dependencies*

**John Mylopoulos**
**University of Toronto**

19th International Conference on Conceptual Modeling
Salt Lake City, October 9-12, 2000




## *Information Models and Conceptual Models*

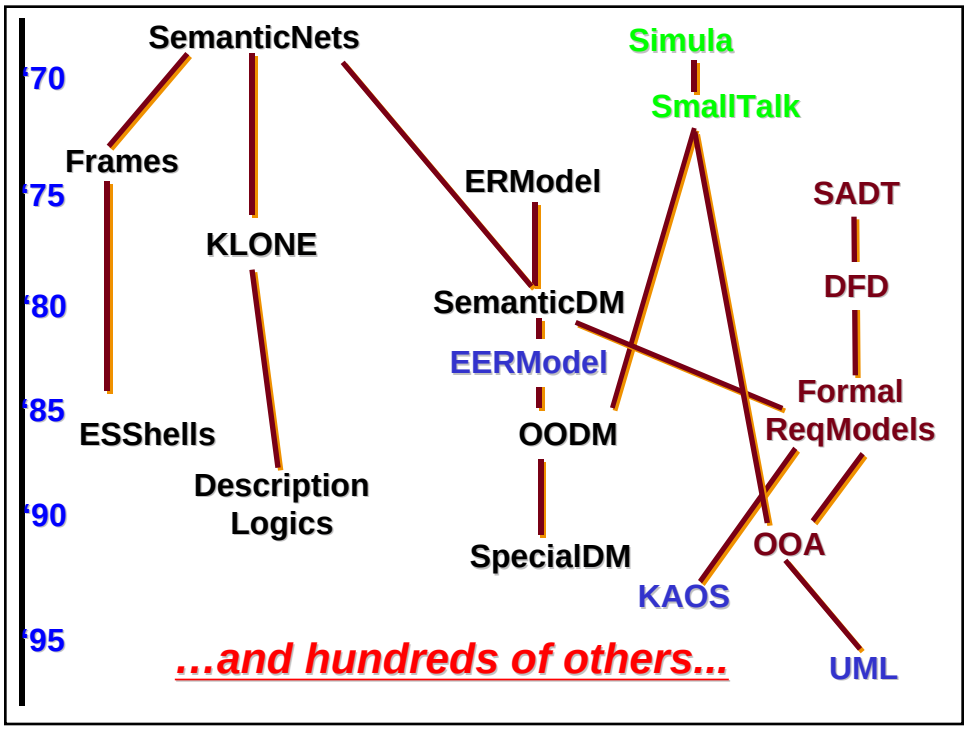
- Information models offer primitives for modeling (information about) an application.
- Physical (information) models offer computer-inspired primitives, such as **`file`**, **`record`**, **`pointer`**, …
- Logical models offer mathematical abstractions, such as **`set`**, **`function`** and **`relation`**.
- Unlike their physical and logical cousins, conceptual models offer:
  - ✓ Primitive concepts which reflect the application being modelled, e.g., **`entity`**, **`relationship`**, **`activity`**, **`agent`**,…
  - ✓ Abstraction mechanisms inspired by Cognitive Science for organizing information, e.g., **`generalization`**, **`aggregation`**, **`classification`**, …

## Origins of Conceptual Modeling

- Ross Quillian proposed in his PhD thesis **semantic networks** in order to model the structure of human memory (1966)
- Ole-Johan Dahl proposed in 1967 **Simula**, an extension of the programming language ALGOL 60, for simulation applications which require some "world modeling"
- Jean-Robert Abrial proposed the **Semantic Model** in 1974, shortly followed by Peter Chen's **Entity-Relationship Model** (1975) as advances over logical data models, such as Codd's Relational Model proposed only a few years earlier.
- Doug Ross proposed in the mid-70s the **Structured Analysis and Design Technique** (**SADT**) as a "language for communicating ideas". The technique was used by Softech, a Boston-based company, in order to model requirements for software systems.

[Mylopoulos98]



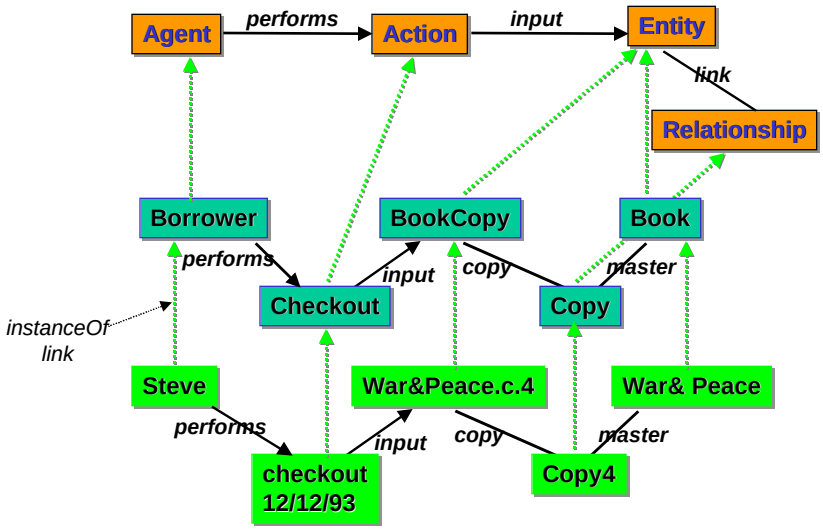
# KAOS

- A formal requirements modeling language.
- Its basic premise: (Organizational) goals lead to requirements.
- Goals justify and explain the presence of requirements which are not necessarily comprehensible by clients.
- Goals can be used to assign responsibilities to agents so that prescribed constraints can be met.
- Goals provide basic information for detecting and resolving conflicts that arise from multiple viewpoints

[KAOS]




# The KAOS Metamodel



## Entities and Relationships in KAOS

**Entity** Library
**Has** collection, available, checkedOut, lost: setOf[BookCopy]
coverageArea: setOf[Subject]
**Invariant** ( collection = available checkedOut lost
available ∩ checkedOut = available ∩ lost =
checkedOut ∩ lost = )
...
**end** Library

**Relationship** Borrowing
**Links** Borrower [**Role** Borrows, Card 0::N]
BookCopy [**Role** BorrowedBy, Card 0::1]
**Invariant** ( lib: Library, bor: Borrower, bc: BookCopy)
(Borrowing(bor,bc) bc lib.collection
bc lib.checkedOut ★Requesting(bor,bc)]
...
**end** Borrowing




## A Coarse-Grain Classification of Primitive Terms

- **Static terms** encompass static aspects of an application, described in terms of concepts such as **`Entity`**, **`Attribute`**, **`Relationship`**, **`Resource`**,...
- **Dynamic terms** model dynamic aspects within an application, described in terms of **`Process`**, **`Activity`**, **`Action`**, **`Plan`**, **`Procedure`**, **`Event`**,...or **`State`**, **`Transition`**,...
- **Intentional terms** describe the world of things agents (human or otherwise) believe in, want, prove, argue about, e.g., **`Issue`**, **`Goal`**, **`Softgoal`**, **`Supports`**, **`Denies`**, **`SubgoalOf`**, ...
- **Social terms** describe social settings in terms of social relationships among agents, such as **`Authority`**, **`Commitment`**, **`Responsibility`**, **`Actor`**, **`Position`**, **`Role`**, **`Social Dependency`**,...
- ...Others...

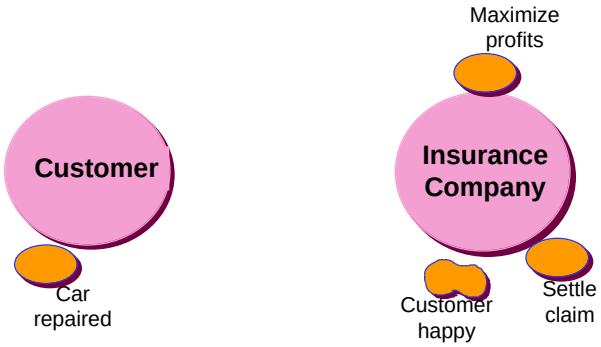
## Intentional and Social Models

- Have been used in AI for planning, intelligent agents, multi-agent systems.
- Have been used, more recently, in Software Engineering for early requirements analysis, also capturing design rationale.
- Are being explored for applications in Workflow and Organizational Modeling, Analysis and Design.
- Have a bright future in Software Engineering.

***Shift from what a software system does…***
***To why it does it, and in cooperation with whom!***

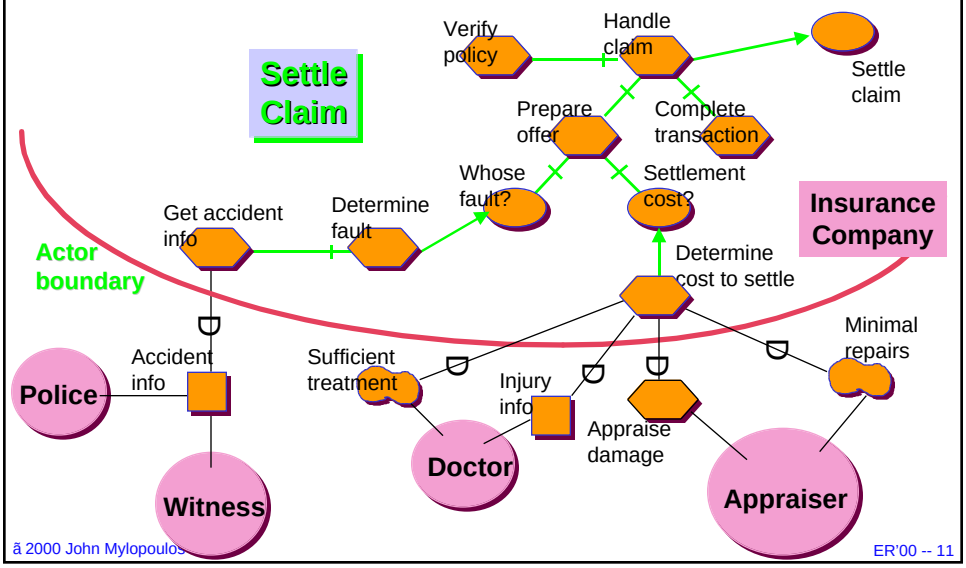



## The i* Framework




***Goals are relative, fulfillment is collaborative***

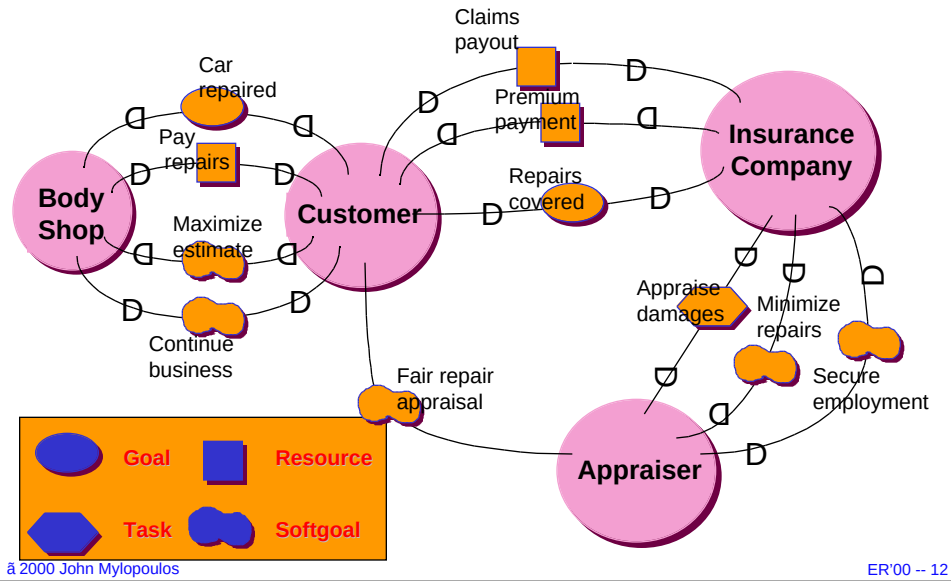
## Means-Ends Analysis







## Strategic Dependency Models



## *Tropos: A Formal Version of i**

- Each concept in an *i** diagram is defined formally, in terms of a KAOS-like specification.
- The specification language includes a temporal logic inspired by KAOS.
- Actors, goals, actions, entities, relationships are described statically and dynamically.




## *A Tropos Example*

**Entity** Claim
**Has** claimId: Number, insP: InsPolicy, claimDate, date: Date, details: Text
**Necessary** date ≤ insP.expDate
**Necessary** ( x)(Claim(x) ●¬Claim(x) ¬RunsOK(x.insP.car))
**end** Claim

**Action** MakeRepair
**Performed by** BodyShop
**Refines** RepairCar
**Input** cl : Claim
**Pre** ¬RunsOK(cl.insP.car)
**Post** RunsOK(cl.insP.car)

## A Goal Dependency Example

**GoalDependency** RepairsCovered
  **Mode** Fulfil
  **Depender** Customer
  **Dependee** InsuranceCo
  **Has** cl: Claim
  **Defined** /* the amount paid out by the insurance company covers repair costs */
**end** RepairsCovered




## Tropos: An Application

- Traditionally, software development techniques have been implementation-driven.
- This means that the programming paradigm of the day dictated the design and requirements paradigms.
- So, structured programming led to structured design and structured (requirements) analysis, while object-oriented programming led to object-oriented design and analysis.
- Aligning the paradigms used for requirements, design and implementation makes perfect sense. But why start with an implementation paradigm?

***What would requirements-driven software development look like??***

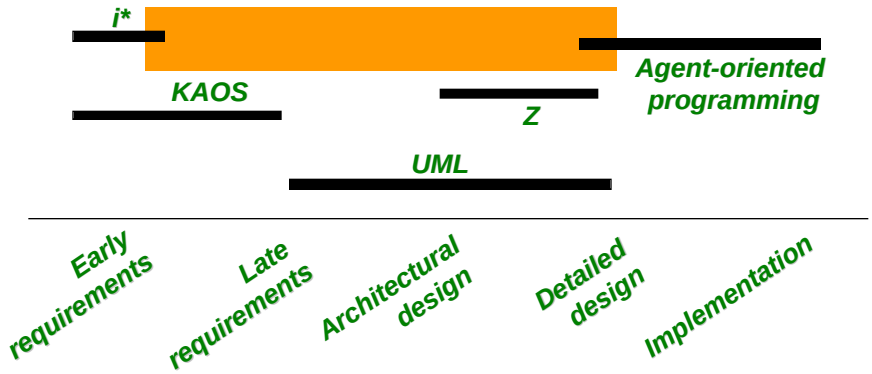
# Agent-Oriented Software Development

- Use the primitive concepts of Tropos to specify early and late requirements, architectural and detailed design for a software system.
- Through all these phases, software is viewed as a social structure whose components are actors (agents, positions or roles) having goals and capable of fulfilling goals, carrying out tasks and delivering resources.
- Unlike other development techniques, goals can be associated with actors in any phase, even at run-time.
- This software development technique is appropriate for software that will operate in unpredictable or underspecified environments, e.g., the internet.

[Tropos]




# Tropos and the Big Picture



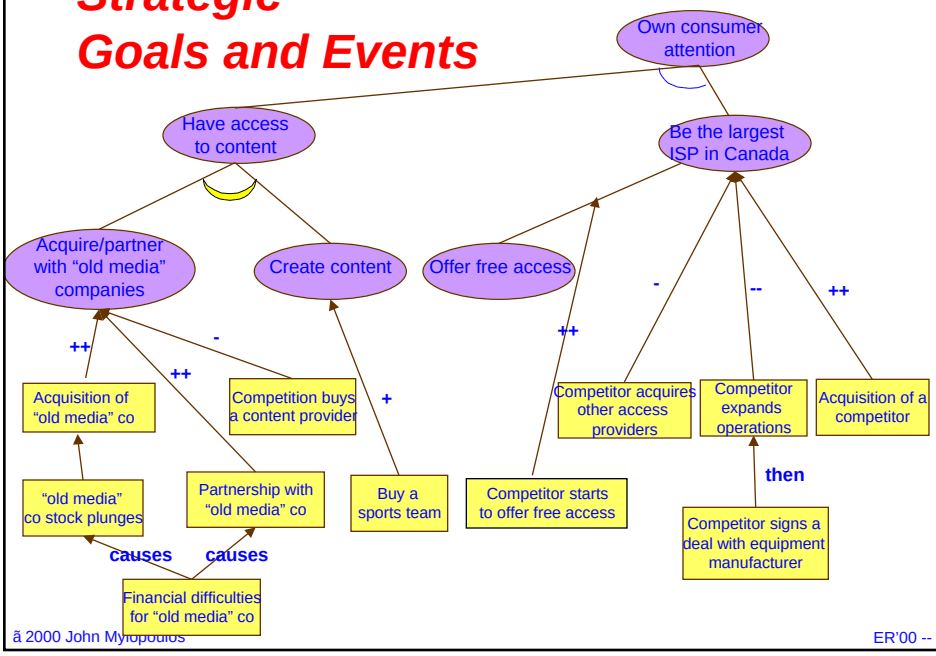
# A Different Application...

- Build a tool that helps strategic business analysts conduct their work.
- Strategic business analysts keep track of trends and events that may affect positively or negatively the strategic objectives of their organization. “Keeping track” includes following (business) news, analysis reports and other relevant information.
- Our solution: Build a model of strategic goals, relevant events, actors etc.

***...If you are holding a hammer,***
***the whole world looks like a bunch of nails...***




# Strategic Goals and Events



## Analysing Models

- Models are used primarily for human communication
- But, this is not enough! Large models can be hard to understand, or take seriously!
- We need analysis techniques which offer evidence that a model makes sense. What would these be for intentional and social models?
  - ✓ Social analysis techniques which look at viability, workability,… for a set of social dependencies.
  - ✓ Goal analysis techniques which determine the fulfillment of a goal, given information about the fulfillment of related goals.
  - → Time simulation which explores the properties of goals over their lifetime.




## Model Checking

- Automatic technique for analyzing finite state concurrent systems.
- Given a model of a specification, represented as a graph, check to see whether it satisfies certain temporal properties.
- Advantages
  - ✓ "push-button" analysis.
  - ✓ always produces a counterexample of properties not satisfied by the model, if one exists (..with some caveats…).

[Clarke99]

## Model Checking for Tropos

- Goal: Apply model checking to richer models than those that have been tried before.
- Approach
  - ✓ Definition of an automatic translation from Tropos specifications to the input language of the nuSMV model checker [Cimatti99].
  - ✓ Verification of temporal properties of state representations of finite Tropos models.
  - ✓ Discovery of interesting scenarios that represent counterexamples to properties not satisfied by the models.
  - ✓ Model simulation.




## Translation for CoverRepairs

```
VAR  CoverRepairs : {no, created, fulfilled}
INIT  CoverRepairs = no
TRANS   CoverRepairs = no -> (next(CoverRepairs)=no |
   next(CoverRepairs)=created)
TRANS   CoverRepairs = created -> (next(CoverRepairs)=created |
   next(CoverRepairs)=fulfilled)
TRANS   CoverRepairs = fulfilled -> next(CoverRepairs) = fulfilled
TRANS    CoverRepairs=no -> next(CoverRepairs = created ->
   !RunOK)
TRANS   CoverRepairs = created -> next(CoverRepairs = fulfilled
   -> DamageCosts = fulfilled)
TRANS   CoverRepairs = created -> next(CoverRepairs = fulfilled
   <-> RunsOK)
```

## An Interesting Property

LTLSPEC F(CoverRepairs = fulfilled) -> F(MakeRepair)

This property does not hold for the model. A counterexample is:

| Variable | $t_1$ | $t_2$ | $t_3$ | $t_4$ |
|---|---|---|---|---|
| RunOK | false | false | true | true |
| DamageCosts | no | no | created | fulfilled |
| CoverRepairs | no | created | created | fulfilled |
| MakeRepair | false | false | false | false |




## A Fix

Add to the definition of the entity class Car

…

**Necessary**

¬ RunsOK(self) ¬ MakeRepair(self) ● ¬ RunsOK(self)

...

## What Comes Next? Metadata!!

- So, you have a zillion data sources out there and want to access them (with computer help). How do you do it??
  Have a ***semantic*** and ***pragmatic*** description of what each contains!
- Prerequisite: Standard models for static, dynamic, intentional, social applications, toolsets that support the process of constructing such descriptions.
- Corequisite: A data model for metadata which supports a rich algebra for model chunks, a "meta" (classification) hierarchy, a rich set of semantic referential relationships, and a first-class treatment for attributes.
- Telos [Mylopoulos90] supports some of these features; we are beginning to work on a data model that supports them all.




## Postscript

- Making conceptual modeling everyday practice, requires more than new ideas and research results.
- Conceptual modeling is only taught in a superficial and "here is another application" manner in most undergraduate curricula for Computer and Information Science.
- Compare how much training our undergraduates get on programming and on modeling!
- Undegraduate curricula need to recognize the importance of modeling in the career of an IT professional.
- Undergraduate curricula need to make sure that students get a solid education on modeling principles and practice!

## *Acknowledgements*

Contributors to the Tropos project include:

**Ariel Fuxman, Manuel Kolp and Eric Yu**
**(University of Toronto, Canada),**
**Jaelson Castro**
**(Federal University of Pernambuco, Brazil),**
**Paolo Bresciani, Paolo Giorgini, Fausto Gunchiglia,**
**Anna Perini, Marco Pistore**
**(University of Trento and IRST, Italy).**




## *The Tropos URL…*

*http://www.cs.toronto.edu/~mkolp/tropos*

## *My email…*

*jm@cs.toronto.edu*

# References


- [Cimatti99] A. Cimatti, E. Clarke, F. Giunchiglia and M. Roveri, "NuSMV: A New Symbolic Model Verifier," in N. Halbwachs and D. Peled, (eds.) Proceedings of the International Conference on Computer-Aided Verification (CAV'99), Trento, Italy, July 1999; Lecture Notes in Computer Science, no. 1633, 495-499, Springer-Verlag.
- [Clarke99] E. M. Clarke, O. Grumberg and D. Peled. *Model Checking,* MIT Press, 1999.
- [KAOS] http://www.ingi.ucl.ac.be/research/projects/AVL/ReqEng.html.
- [Mylopoulos90] J. Mylopoulos, A. Borgida, M. Jarke, and M. Koubarakis. "Telos: Representing Knowledge About Information Systems," *ACM Transactions on Information Systems*, 1990.
- [Mylopoulos98] J. Mylopoulos, "Information Modeling in the Time of the Revolution", *Information Systems 23(3-4)*, June 1998.
- [Tropos] http://www.cs.toronto.edu/~mkolp/tropos





# Information Models