

# ***Tropos at the Age of 10 months***

**John Mylopoulos  
University of Toronto**

**Tropos Meeting  
November 13, 2000**

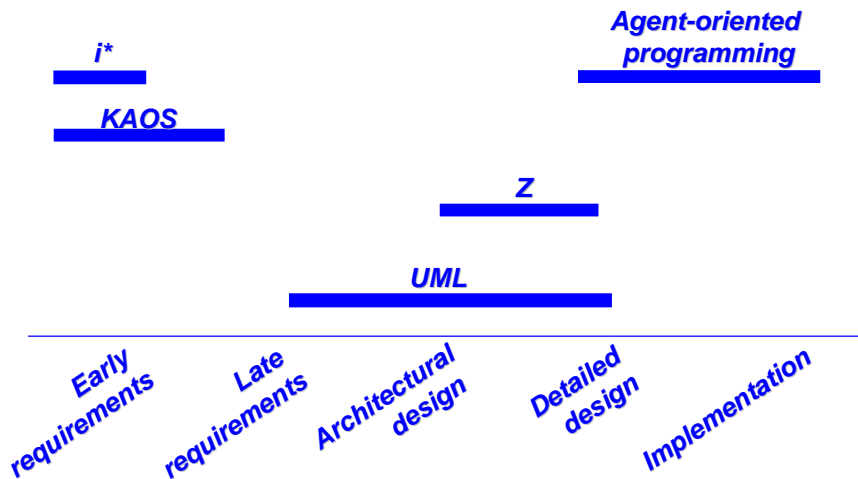


## ***Tropos\****

- Ontology** Actors, social dependencies, goals, resources,...
- Structuring** Contexts (+generalization, aggregation, classification and attribution)
- Application area** (Requirements-driven) Software development
- Tools** Software development, process analysis, enactment support, goal and softgoal analysis

\* **Tropos**, in Greek, means **manner** (as in "manner of doing things")

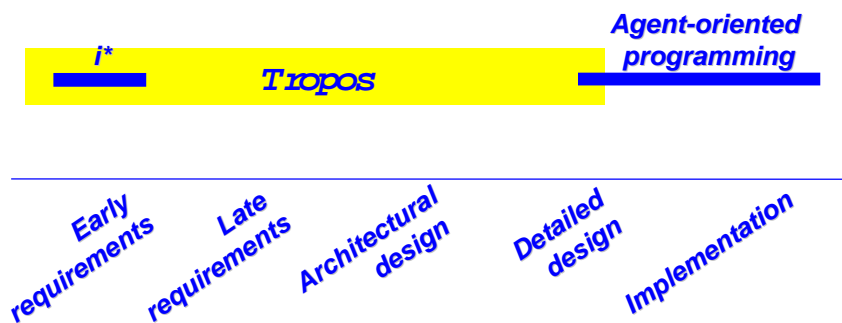
## Where Are We??



© 2000 John Mylopoulos

Tropos -- 3

## Where Do We Want To Be??



**Guiding Principle: Push concepts as far down as possible (...and see what happens!)**

© 2000 John Mylopoulos

Tropos -- 4

## **Our Research Baseline**

- People: Jaelson Castro, Ariel Fuxman, Manuel Kolp, Eric Yu (University of Toronto); Paolo Bresciani, Paolo Giorgini, Fausto Giunchiglia, Anna Perini, Marco Pistore, Paolo Traverso (University of Trento/IRST)
- Our baseline is defined by  $i^*$ , nuSMV, but also CONGOLOG.
- nuSMV is IRST's extension of the SMV model checker.
- CONGOLOG offers a logic-based framework for modelling actions (but also activities, processes, goals, and agents)
- We hope to use nuSMV and CONGOLOG in order to support different forms of formal analysis for Tropos specifications.

## **A Multi-Perspective View of Software**

- We are working towards an agent-oriented software development methodology, founded on the key concepts of actor, goal, (goal, task, resource) dependency, etc.
- Software is viewed from four perspectives:
  - **Organizational** -- who are the relevant actors, what do they want? What are their obligations? ...capabilities??
  - **Intentional** -- what are the relevant goals and how they interrelate? How are they being met? ... by whom??
  - **Process-oriented** -- what are the relevant business/computer processes? Who is responsible for what?
  - **Object-oriented** -- relevant objects
- We are focusing on organizational and intentional perspectives because they are novel.

## ***From Diagrams to Formal Specs***

- Diagrams are not complete nor formal as software specifications.
- We propose to offer three levels of software modelling:
  - Diagrams, as proposed in *i\**.
  - Partially formal annotations, to complement diagrammatic notations, e.g., annotations may specify that some obligation takes precedence over another
  - Formal specs, using some form of logic, which are amenable to analysis
- Diagrams are great for communication, partially formal annotations can help in defining some forms of analysis, formal specs can serve as foundation for a range of analysis techniques, including proofs of correctness, process simulation, goal analysis etc.
- We propose to use UML-type modelling techniques, where possible, at the diagrammatic level.

## ***Research Issues***

- An agent-oriented software development methodology with clear guidelines on what decisions are made when.
- Applications of planning paradigms to agent-oriented software development.
- Modeling (multi-agent) processes with coordination and exceptions.
- Simulation and invariant analysis for agent and goal models.
- Viability and workability analysis for dependencies.
- Filling and managing positions and roles.
- Goal and softgoal means-ends analysis.
- Exception handling (**note:** this should be interesting)
- Management of commitments and obligations, individual and global issues.

## ***Research Tasks***

- Case studies -- doing early and late requirements, architectural and detailed design in Tropos;
- Modelling issues for late requirements, architectural design, detailed design: diagrams, annotations, formal specs;
- Forms of analysis at any of the 4x3 levels of specification;
- Language design for Tropos;
- An environment for building and analyzing Tropos specs;
- ...more...