

# Agent Design Methodologies: What, How, Tools, and Issues

Michael Winikoff and Lin Padgham  
RMIT University

Workshop on Agent Oriented Software Engineering  
4 November 2002,  
Seattle

(Assume you've read the papers!)

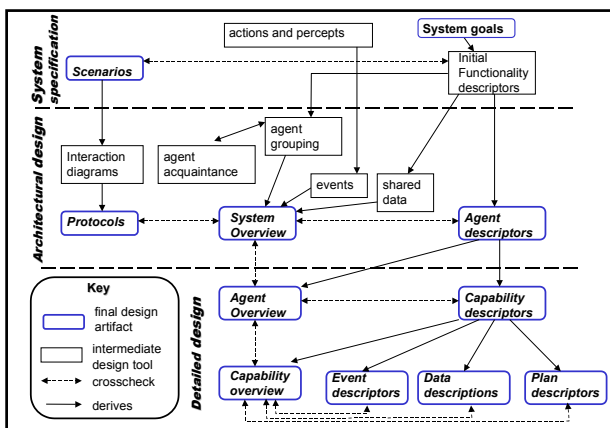


## Prometheus' Goals

- Support development of *intelligent* agents
- Provide "start-to-end" support and a *detailed process*
- Evolve out of practical industrial and pedagogical experience
- Provide hierarchical structuring
- Use an iterative process – not waterfall
- Provide tool support: especially automated cross checking



Prometheus was the wisest Titan. His name means "forethought" and he was able to foretell the future. Prometheus is known as the protector and benefactor of man. He gave mankind a number of gifts including fire.  
<http://www.greekmythology.com/>



## Notes on Previous Slide

- Shows models and dependencies
- Doesn't show process (but implies it)
- Doesn't show implicit iteration
- Separates design phase into architectural and detailed design - only latter depends on the implementation platform.

## Agent **Design** Methodology

- To formulate a plan for; devise. To make or execute plans.
- To **plan** out in systematic, **usually graphic form**: design a building; design a computer program.
- To create or contrive for a **particular purpose or effect**: a game designed to appeal to all ages.
- **To create designs**
- A graphic representation, especially a detailed plan for construction or manufacture.
- **The art or practice of designing or making designs**
- A basic scheme or pattern that affects and controls function or development

## Some Design Questions

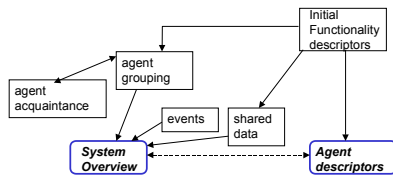
- What agent types ("classes") should there be?
- What agents ("objects") should there be?
- What is the structure of the system?
- How does the system behave?
- How do agents achieve their goals?
- How do agents respond to events?
- What do agents need to know?



→ **These are the sorts of questions that design should address!**

### What Agent Types?

- Identify agents by grouping functionalities.
- Grouping based on coupling & cohesion.
- Investigate alternative designs



### Agent Design **Methodology**

- A body of practices, procedures, and rules used by those who work in a discipline or engage in an inquiry; a set of working methods.
- Usage Problem. Means, technique, or procedure; method.

*Just process?*

### **Agent** Design Methodology

- Agent:
  - Situated in an environment
  - Autonomous – operates without supervision
  - Proactive – works to achieve its goals
  - Reactive – responds (in a timely manner) to changes in the environment
  - Social – interacts with other agents (at a suitable level)



- Drives concepts
- Defines scope of methodology, e.g. agents are not in open systems, are not mobile, are not in teams.

### Concepts

(based on notion of agent)

- Agent, Environment
- Actions
- Percept: Incoming information from the environment
- Incident: A significant occurrence, internal/external (event)
- Goals (conditions to be achieved)
- Data (can be beliefs, objects, etc.)
- Plans (triggering goal, context, body)

Michael Winikoff, Lin Padgham, and James Harland. **Simplifying the Development of Intelligent Agents**. In AI2001: Advances in Artificial Intelligence. 14th Australian Joint Conference on Artificial Intelligence. LNAI 2256, pages 557-568, Adelaide, December 2001.

### Issues

- General
- Pragmatics
- Modelling & Notation
- Tool Support
- UML

### General Issues

- Generality vs. usefulness
- Academic novelty vs. usefulness
  - E.g. adding Tropos' early requirements to Prometheus
- Industrial Adoption?
  - Agents first, then methodologies
- Formality vs. usability/accessibility
  - But need to have precise definition, even if only used by tool-builders!
- Development contexts: don't assume "green field"

## General Issues


- Evaluation!
  - Many methodologies – how to decide which are good? How to choose which to use?
- “Non-classical” phases: debugging, deployment
- Maintenance phase: iteration
  - Traceability!
  - Tool support
- What’s your view of what an agent is?
- Concurrency!
- Extending to open systems, teamwork, mobile agents, security
- Validation of methodologies

## Pragmatics

- Legacy software integration
- Cost (training, tools, ...)
- System performance
- Training and documentation availability
- Tool support ...



## Modelling/Notation Issues

- Graphical notation vs. text
- Precise definition of notation syntax & semantics - Is  valid?
- Allow different views/slices/aspects of system
- Static
- Dynamic
- Link notation and concepts
- Notation: mnemonic, easy to draw (hand/tools), print etc.
- Interchange of graphical models

## Tool Support

- Whiteboard
- Word processors
- Diagram editors (general/specialised)
- Consistency checker
- Process modelling
- Automated testing
- Rapid prototyping
- Code generator
- Debugger
- Performance analysis




## UML

- Notation and concepts, no process
- Aimed at objects – thus partially relevant to agents
- But there are differences:
  - Treating messages as entities, not labels
  - Percept/action/message, and percept processing
  - Agents are active, beliefs are passive
- ...

## Some Current Work at RMIT

- Protocol representation and use (in detailed design)
- Concurrent goal management (John Thangarajah)
- Debugging techniques utilizing design artifacts (David Poutakidis)
- Testing techniques (e.g. using scenarios)
- Evaluating AOSE Methodologies (Khanh Hoa Dam)
- Open Systems

## Questions?



Copyright © 1999 United Feature Syndicate, Inc.  
Reproduction in whole or in part prohibited

<http://www.cs.rmit.edu.au/agents/SAC>  
 → <http://www.cs.rmit.edu.au/agents> ←  
<http://www.cs.rmit.edu.au/~winikoff>  
[winikoff@cs.rmit.edu.au](mailto:winikoff@cs.rmit.edu.au)

## Why not Existing Methodologies?

- High level design differs for different programming paradigms, different abstractions:
  - Procedural: What does it do?
  - OO: What objects are there? (data+operations)
  - Agent: What goals are there? What are the relationships between agents?
- Low level design differs since agent systems face uncertainty and failing actions
  - Need to have alternative plans - can't assume things will work
- Knowledge level: beliefs, goals, etc.