

# AOSE regarded as a software methodology for the real-world

- Agent: **autonomous**, **situated** in an environment
- Real world: dynamic, uncertain, non-deterministic ...things *will* go wrong
- An intelligent agent blends **pro-active** & **reactive** behaviour:



- Also flexible, robust, rational, social, ...

# Agents vs objects

In the object-oriented approach

- objects have identity, **state** and **behaviour** and communicate via messages

In the agent-oriented approach

- Agents have identity, **state** (knowledge, beliefs, desires, intentions) and **behaviour** (plans to achieve goals, actions, reactions to events)

So are objects agents?

- Agents exhibit autonomy, they have control over their state, execution and ultimately behaviour
- Agents exhibit goal-directed, reactive and social behaviour
- They are persistent, self-aware and able to learn and adapt
- Agents have their own thread of control in a multi-agent system
- Control in multi-agent systems is also distributed

# Agent Oriented Software Engineering (AOSE)

- Software Engineering ...
  - “We know how to do SE, how apply to agents?”
- ... of Agent Oriented systems
  - “We know about agents and AI, how engineer large (agent) systems?”
- Relatively recent development (first AOSE workshop was in 2000)



# Why not traditional SE?

- 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?

# Why not traditional SE?

- Low level design differs since agent systems face uncertainty and failing actions:
  - Need to have alternative plans - can't assume things will work
  - Not monolithic single plans!

# Agent Concept Soup *team*

goal      **obligation**      *task*      *behaviour*      **desire**

*percept*      **protocol**      *time*      **belief**      *commitment*      *role*

**intention**      *action*      *situation*

Knowledge base      *resource*      **event**

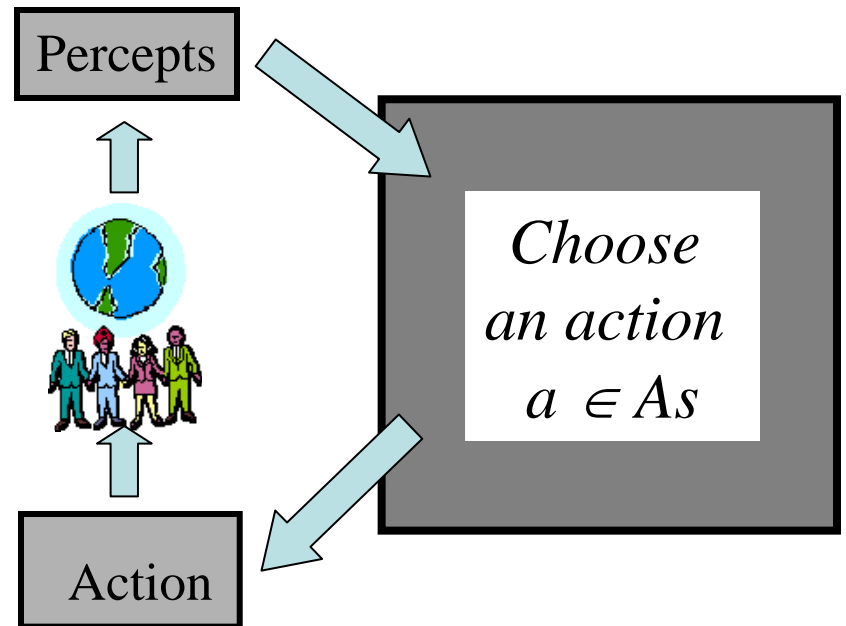
plan      *capability*      *aim*

**choice**      *state*      **decision**      *world*

*objective*

# Agent Concepts

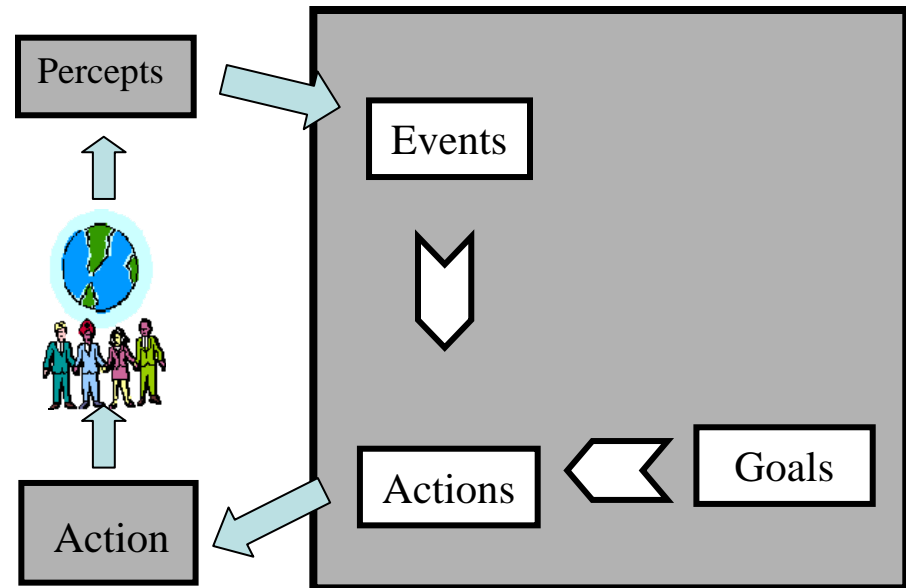
- **Situated** so **actions**, **percepts**, time
- Fire engine:
  - See nearby fires & road conditions, hear messages from other agents, hear civilian calls for help.
  - Move, squirt, tell (broadcast), say, plan route (internal)





# Agent Concepts

- **Reactive** so **events**  
(significant occurrence)
  - *New fire, fire extinguished, fire urgent, help requested*
- **Proactive** so **goals**
  - *Put out fire, discover fire, assist, coordinate*



# Agent Concepts

- **Implementation** uses **plans** and **beliefs**

- Cache for means, and world information respectively

- *Beliefs: Map (incl. fires, buildings), fire assignment and priority*

- *Plans: Put out fire, roam, ...*

