COMP9006 Knowledge Representation

Introductory Lecture

Dr Ruairi O'Reilly

September 17, 2019

Cork Institute of Technology

My Background - Teaching

Teaching Experience

- 10+ years Teaching Experience
- Developed the MSc in AI
- Developed the MSc in DA
- Online Programme Manager

Modules

- COMP9016 Knowledge Representation
- COMP8054 Interactive Data Visualisation

My Background - Research

Prior Projects

- The BabyLink Project (PhD - SFI)
- ScrutiniseIT (Researcher - EI)
- Cloud Lightning (Researcher - H2020)
- INFANT (Researcher - SFI)

Current Projects

- Beats-Per-Minute (CIT/ADVANCE CRT) PhD Student
- SMART LMI (ERASMUS+)
- Artificially Intelligent energy and buildings data Mining System (AIMS)

My Background - Research Supervision

Phd Students

- Ryan Donovan - Effective Computing
- Urja Pawar - Personalised Healthcare

MSc Students

- Orla O’Brien BPM (HealthCom2018)
- David Foley (AICS2018)

Contact Details

- **E-mail**: ruairi.oreilly@cit.ie
- **General problems**: Interact with one another in person or online via Canvas.
- **Personal problems**: Issues related to module then e-mail me.
- **Office hours**: By appointment only.

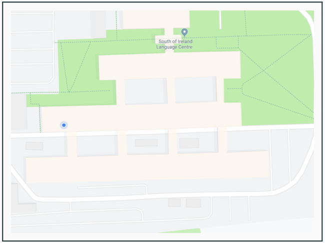


Figure 1: My Office - B180A

COMP9016 - Module Overview

COMP9016 Learning Outcomes

LO1 Appraise domain specific formalisms used in knowledge representation schemes.

LO2 Compare and contrast current knowledge representation approaches integrated in systems relevant to AI.

LO3 Select, apply and evaluate a knowledge representation scheme for a specified domain.

LO4 Design and implement KR formalisms for a real world data set.

LO5 Interpret, critique and communicate the suitability of data visualisation techniques used in conjunction with the design of KR formalisms and the analysis of the resulting output.

What is Knowledge Representation?

- A field of AI that focuses on the representation of domain specific knowledge in a form that can be utilised by computer systems.
- KRs within a domain are often conceived as formalisms.
- A formalism??? A description of something in formal mathematical or logical terms.

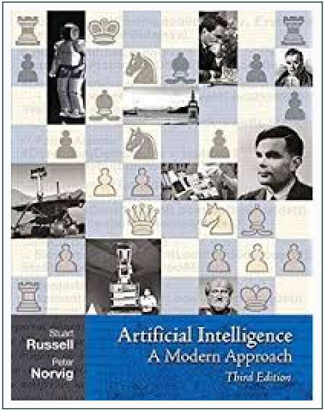
What I expect you to take away from Knowledge Representation (KR) and Reasoning

- Methodologies for the visualisation and interpretation of domain specific knowledge
- The ability to translate data interpretations into KR formalisms.
- An appreciation of how to evaluate the suitability of knowledge representation schemes, balance competing features/requirements and make informed decisions when designing KR formalisms.
- Practical experience of applying KR to appropriate real world problems such as the semantic web, time-series indexing and temporal abstraction of expert knowledge.

Recommended Reading

“Artificial Intelligence A Modern Approach 3rd Edition” Stuart Russell, Peter Norvig

Buy it, start reading and studying it!



Assessment 100% CA

- **Project (40%)**: Detail and critically evaluate a knowledge representation scheme, its utilisation of formalisms and relevance to Artificial Intelligence. (W6)
- **Project (60%)**: Design, develop and deploy a KR solution for a real world problem domain. e.g. a system for automating the interpretation of biomedical data requires a KR scheme that enables the translation of data such that the integrity of all domain specific information is maintained. Provide a rationale for the chosen visualisation approach taken at both the design and analysis stage. (Semester End)

Your Level of Knowledge

- You can program?
- Technical Level 8 qualification? (Computer Science, Engineering, Mathematics etc.)
- How much AI have you have covered previously?

What we will cover

Part I Artificial Intelligence - (Intro to AI & Intelligent Agents)

Part II Problem Solving - (Problem-Solving Agents, Example Problems, Searching for Solutions)

Part III Knowledge and Reasoning - (Logical Agents, First Order Logic, Inference in First-Order Logic, Knowledge Representation)

Part IV Uncertain Knowledge and Reasoning - (Quantifying Uncertainty, Probabilistic Reasoning, Probabilistic Reasoning over Time, Making Simple Decisions, Making Complex Decisions)

Part V Learning - (Learning from Examples, Knowledge in Learning, Learning Probabilistic Models, Reinforcement Learning)

What we will cover - Applied Work

Feature Engineering - (Introduced via lab & assignments, short focused problems that are self directed, quick turn-around times.

Time Series Analysis - (Introduced via lab assignments and research papers/case studies - time series indexing, motif discovery, representation etc.)

Data Visualisation - (Notes Analytical and Scientific Programming, theory from Interactive Data Visualisation)

Artificial Intelligence - Brief Overview

What is AI?

Views of AI fall into four categories:

Systems that think like humans	Systems that think rationally
Systems that act like humans	Systems that act rationally

Examining these, we will push for acting rationally

Acting humanly: The Turing test

Turing (1950) “Computing machinery and intelligence”:
“Can machines think?” $\longrightarrow$ “Can machines behave intelligently?”
Operational test for intelligent behavior: the Imitation Game

Predicted that by 2000, a machine might have a 30% chance of AI fooling a lay person for 5 minutes
Anticipated all major arguments against AI in following 50 years
Suggested major components of AI: knowledge, reasoning, language, understanding, learning

Problem: Turing test is not *reproducible*, *constructive*, or amenable to *mathematical analysis*

Thinking humanly: Cognitive Science

1960s “cognitive revolution”: information-processing psychology replaced prevailing orthodoxy of behaviorism

Requires scientific theories of internal activities of the brain:

- What level of abstraction? “Knowledge” or “circuits”?
- How to validate? Requires
    1. Predicting and testing behavior of human subjects (top-down)
    2. Direct identification from neurological data (bottom-up)

Both approaches (roughly, Cognitive Science and Cognitive Neuroscience) are now distinct from AI.

Thinking rationally: Laws of Thought

Normative (or *prescriptive*) rather than *descriptive*

Aristotle: what are correct arguments/thought processes?

Several Greek schools developed various forms of *logic*:

notation and *rules of derivation* for thoughts;
may or may not have proceeded to the idea of mechanization

Direct line through mathematics and philosophy to modern AI

Problems:
1) Not all intelligent behavior is mediated by logical deliberation
2) What is the purpose of thinking? What thoughts should I have?

Acting rationally

Rational behavior: doing the right thing

The right thing: that which is expected to maximize goal achievement,
given the available information

Doesn't necessarily involve thinking—e.g., blinking reflex—but thinking should be in the service of rational action

Aristotle (Nicomachean Ethics): *Every art and every inquiry, and similarly every action and pursuit, is thought to aim at some good*

Rational agents

An *agent* is an entity that perceives and acts

This course is about designing rational agents

Abstractly, an agent is a function from percept histories to actions:

$$f : \mathcal{P}^* \rightarrow \mathcal{A}$$

For any given class of environments and tasks, we seek the agent (or class of agents) with the best performance

Caveat: computational limitations make perfect rationality unachievable
$\rightarrow$ design best **program** for given machine resources

State of the art - Which of the following can be done at present?

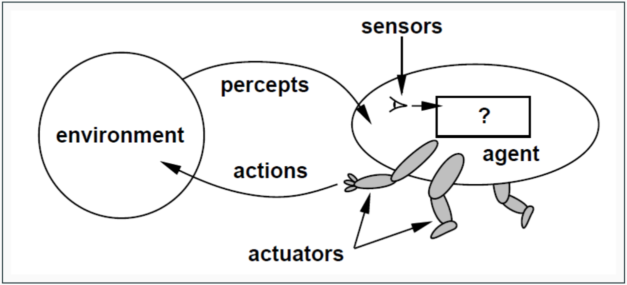
- Play a decent game of table tennis
- Drive safely along a curving mountain road
- Drive safely along Telegraph Avenue
- Buy a week's worth of groceries on the web
- Buy a week's worth of groceries at Berkeley Bowl
- Play a decent game of bridge
- Discover and prove a new mathematical theorem
- Design and execute a research program in molecular biology
- Write an intentionally funny story
- Give competent legal advice in a specialized area of law
- Translate spoken English into spoken Swedish in real time
- Converse successfully with another person for an hour
- Perform a complex surgical operation

Intelligent Agents

Outline

- Agents and environments
- Rationality
- PEAS (Performance measure, Environment, Actuators, Sensors)
- Environment types
- Agent types

Agents and environments



Agents include humans, robots, softbots, thermostats, etc.

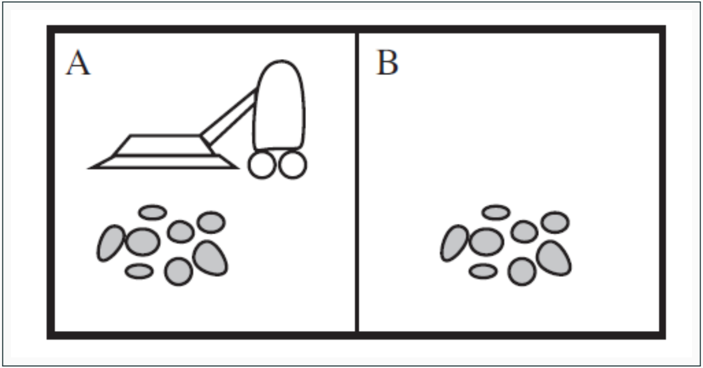
The agent function maps from percept histories to actions:

$$f : \mathcal{P}^* \to \mathcal{A}$$

The agent program runs on the physical architecture to produce:

$$f$$

Vacuum-cleaner world



A vacuum-cleaner agent

Percept Sequence	Action
A : Clean	Right
A : Dirty	Suck
B : Clean	Left
B : Dirty	Suck
A : Clean, A : Clean	Right
A : Clean, A :Dirty	Suck

```
function Reflex-Vacuum-Agent( [location,status]) returns A
        if status = Dirty then return Suck
        else if location = A then return Right
        else if location = B then return Left
```

What is the right function? Can it be implemented in a small agent program?

Rationality

Fixed performance measure evaluates the environment sequence

- one point per square cleaned up in time T?
- one point per clean square per time step, minus one per move?
- penalize for $>$ k dirty squares?

A **rational agent** chooses whichever action maximizes the **expected** value of the performance measure **given the percept sequence to date**

Rational $\neq$ omniscient (percepts may not supply all relevant information) Rational $\neq$ clairvoyant (action outcomes may not be as expected) Hence, rational $\neq$ successful

Rational *implies* exploration, learning, autonomy

PEAS

To design a rational agent, we must specify the **task environment**

Consider, e.g., the task of designing an automated taxi:

- Performance measure??
- Environment??
- Actuators??
- Sensors??

PEAS

To design a rational agent, we must specify the **task environment**

Consider, e.g., the task of designing an automated taxi:

- Performance measure?? safety, destination, profits, legality, comfort, ...
- Environment?? US streets/freeways, traffic, pedestrians, weather, ...
- Actuators?? steering, accelerator, brake, horn, speaker/display, ...
- Sensors?? video, accelerometers, gauges, engine sensors, keyboard, GPS, ...

Internet shopping agent

- Performance measure??
- Environment??
- Actuators??
- Sensors??

Internet shopping agent

- Performance measure?? price, quality, appropriateness, efficiency
- Environment?? current and future WWW sites, vendors, shippers
- Actuators?? display to user, follow URL, ll in form
- Sensors?? HTML pages (text, graphics, scripts)

Environment types

	Solitaire	Backgammon	Online-Shop	Taxi
Observable				
Deterministic				
Episodic				
Static				
Discrete				
Single-agent				

The environment type largely determines the agent design

The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

Environment types

	Solitaire	**Backgammon**	**Online-Shop**	**Taxi**
Observable	Yes	Yes	No	No
Deterministic				
Episodic				
Static				
Discrete				
Single-agent				

The environment type largely determines the agent design

The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

Environment types

	Solitaire	**Backgammon**	**Online-Shop**	**Taxi**
Observable	Yes	Yes	No	No
Deterministic	Yes	No	Partly	No
Episodic				
Static				
Discrete				
Single-agent				

The environment type largely determines the agent design

The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

Environment types

	Solitaire	**Backgammon**	**Online-Shop**	**Taxi**
Observable	Yes	Yes	No	No
Deterministic	Yes	No	Partly	No
Episodic	No	No	No	No
Static				
Discrete				
Single-agent				

The environment type largely determines the agent design

The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

Environment types

	Solitaire	**Backgammon**	**Online-Shop**	**Taxi**
Observable	Yes	Yes	No	No
Deterministic	Yes	No	Partly	No
Episodic	No	No	No	No
Static	Yes	Semi	Semi	No
Discrete				
Single-agent				

The environment type largely determines the agent design

The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

Environment types

	Solitaire	Backgammon	Online-Shop	Taxi
Observable	Yes	Yes	No	No
Deterministic	Yes	No	Partly	No
Episodic	No	No	No	No
Static	Yes	Semi	Semi	No
Discrete	Yes	Yes	Yes	No
Single-agent				

The environment type largely determines the agent design

The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

Environment types

	Solitaire	**Backgammon**	**Online-Shop**	**Taxi**
Observable	Yes	Yes	No	No
Deterministic	Yes	No	Partly	No
Episodic	No	No	No	No
Static	Yes	Semi	Semi	No
Discrete	Yes	Yes	Yes	No
Single-agent	Yes	No	Yes*	No

The environment type largely determines the agent design

The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

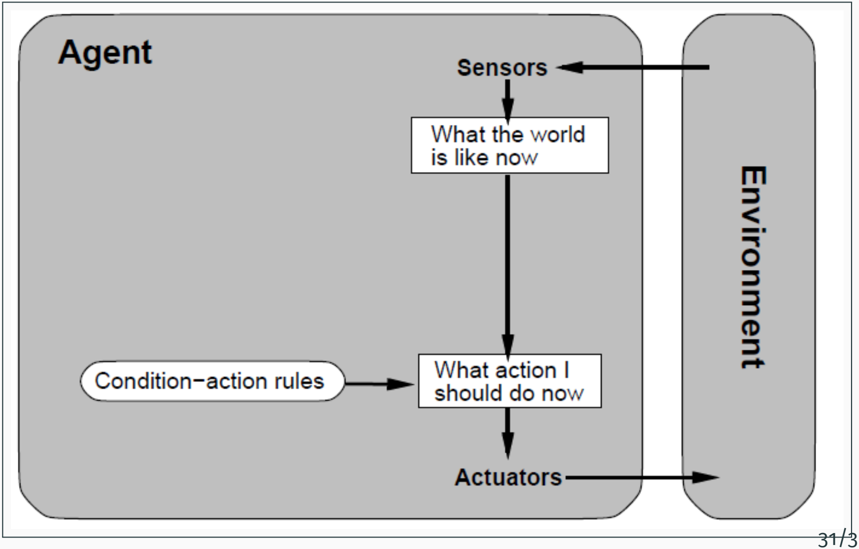
Agent Types

Four basic types in order of increasing generality:

- simple reflex agents
- reflex agents with state
- goal-based agents
- utility-based agents

All these can be turned into learning agents

Simple reflex agents

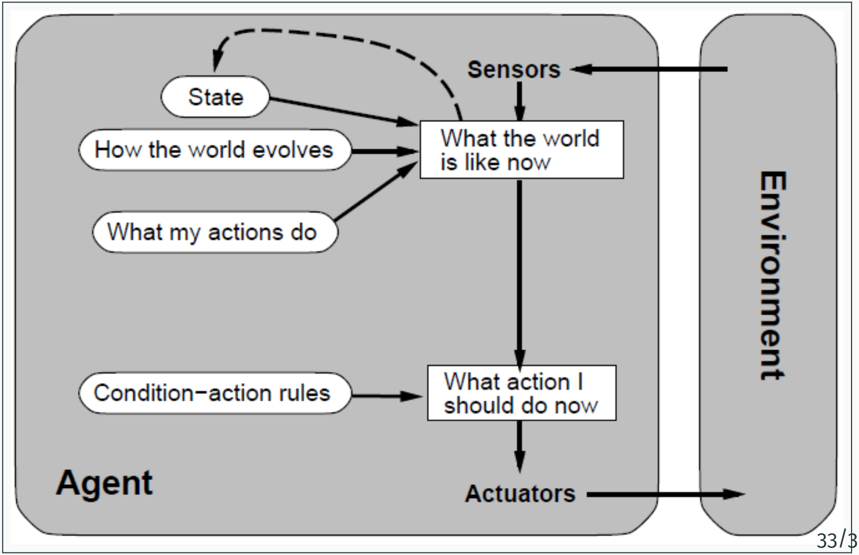


Example

```
function Reflex-Vacuum-Agent( [location,status]) returns A
        if status = Dirty then return Suck
```

```
def REFLEX_VACUUM_AGENT(location, status):
        if status == "Dirty":
                return "Suck"
        elif location == "A":
                return "Right"
        elif location == "B"
                return "Left"
```

Reflex agents with state



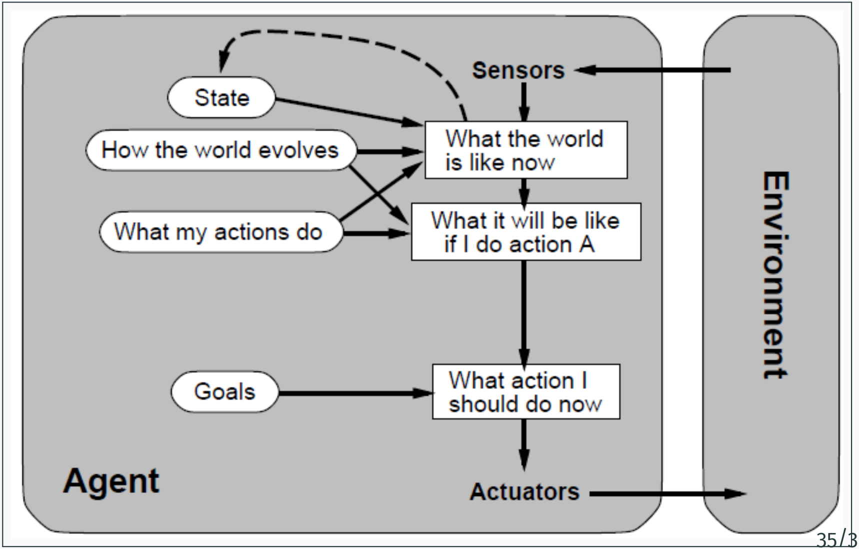
Example

```
function Reflex-Vacuum-Agent( [location,status]) returns A
        static: last_A, last_B
```

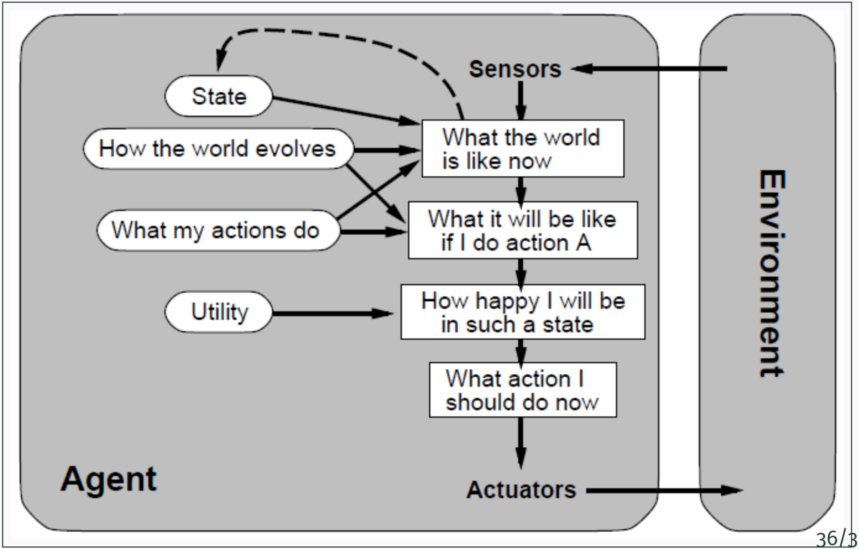
```
def REFLEX_VACUUM_AGENT_WITH_STATE(percept):
        location, status = percept
        state = {loc_A: None, loc_B: None}
        """Same as ReflexVacuumAgent, except if everything is cl

    state[location] = status  # Update the model here
    if state[loc_A] == state[loc_B] == 'Clean':
            return 'NoOp'
        elif status == 'Dirty':
            return 'Suck'
        elif location == loc_A:
            return 'Right'
        elif location == loc_B:
```

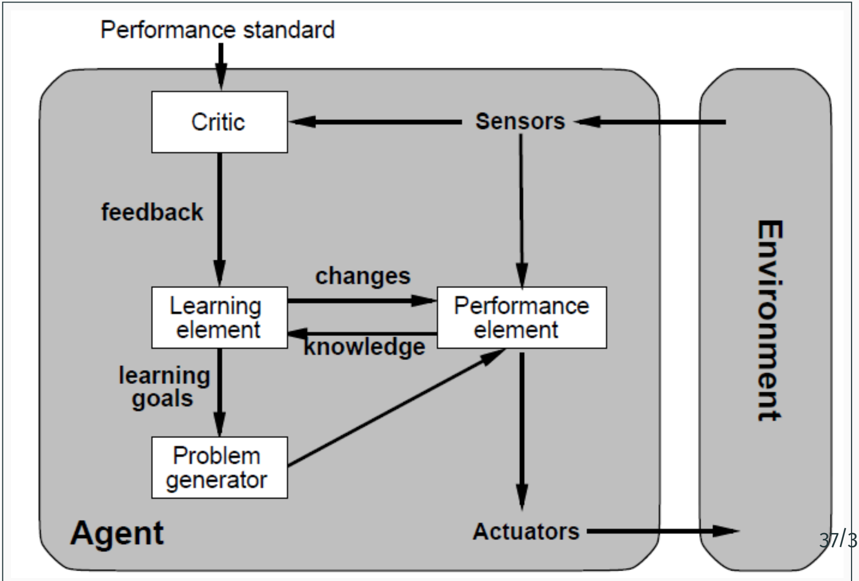
Goal-based agents


Sensors
State
How the world evolves
What my actions do
What the world is like now
What it will be like if I do action A
Goals
What action I should do now
Agent
Actuators
Environment

Utility-based agents


Sensors
State
How the world evolves
What my actions do
Utility
What the world is like now
What it will be like if I do action A
How happy I will be in such a state
What action I should do now
Agent
Actuators
Environment

Learning agents



Summary

Agents interact with environments through actuators and sensors

The agent function describes what the agent does in all circumstances

The performance measure evaluates the environment sequence

A perfectly rational agent maximizes expected performance

Agent programs implement (some) agent functions

PEAS descriptions define task environments

Environments are categorized along several dimensions: observable? deterministic? episodic? static? discrete? single-agent?

Several basic agent architectures exist: reflex, reflex with

Week 1 - To Do

- Order a copy of “AI A Modern Approach 3rd Edition”
- Read Chapter 1 & 2 of “AI A Modern Approach 3rd Edition”
- Download the AIMA python data repo and play around with the agent environment

References

Content & Figures - “Artificial Intelligence: A Modern Approach 3rd Edition”

LaTex slides - some content taken directly from `https://people.eecs.berkeley.edu/~russell/slides/`