CompSci.367

The Practice of Artificial Intelligence

Assoc. Prof. Ian Watson

A Short History of AI

- The origins of Artificial Intelligence
- General Purpose Problem Solvers
- Expert Systems - a solution ?
- The early years
- The hype
- The AI Winter
- The AI Spring

Science Fiction

- we are all familiar with the concept of intelligent machines: *2001, Bladerunner, the Terminator, Star Trek*

Science Fiction

- Movies you should see this semester:
  - *2001* www.youtube.com/watch?v=JcNkMIwolKc
  - *Blade Runner* www.youtube.com/watch?v=J_hYs1jBy8Y
  - *the Terminator* www.youtube.com/watch?v=LLHik9TiJkA
  - *A.I.* www.youtube.com/watch?v=7xeeteKWb6U
  - *Dark Star* www.youtube.com/watch?v=qjGRySVyTDk

The Origins of AI

- In the 1830s - 40s Charles Babbage an English inventor and mathematician designed an "engine" for automating calculation
- His *Difference Engine* was never built in his life time

Babbage's Difference Engine No.2

www.youtube.com/watch?v=0anIyVGeWOI

The Origins of AI

- After the Difference Engine Babbage designed the *Analytical Engine*
- This could be programmed by punch cards to solve "any" mathematical problem
- The Analytical Engine has the same theoretical architecture as a modern digital computer
  - I/O devices, program store, CPU & working memory
  - www.youtube.com/watch?v=GJiyGvoYd5E

The Origins of AI

- Lady Ada Lovelace hypothesised in 1842 that Charles Babbage's Analytical Engine could manipulate symbols other than numbers and hence perhaps could compose music or poems
- The programming language ADA is named after her

http://en.wikipedia.org/wiki/Ada_lovelace

The Frankenstein myth

- An enduring myth
- Brought to life by Mary Shelly in her gothic novella *Frankenstein*
- echoed in 2001, Terminator, etc….
- Interesting relationship to the origins of computing & AI

The Frankenstein myth

- Mary Shelley was married to Percy Shelly
- Best friend of Lord George Byron
- Ada Countess of Lovelace was Byron's daughter
- Ada worked with Charles Babbage
- and hypothesized about AI

The Frankenstein myth

- Ada must have read Frankenstein
- The idea of creating a conscious entity that may turn upon us was already in popular culture around 150 years ago
- At the birth of computing & AI

www.sdsc.edu/ScienceWomen/lovelace.html

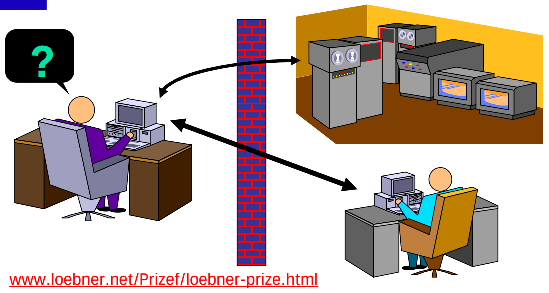
The Origins of AI

- Alan Turing showed that a very simple computer (a Turing machine) could solve any problem that could be described by symbols
- In the 1930's he is the person first credited with proposing that a computer could exhibit "intelligence"
- During WW-2 he worked cracking German codes
- He worked on the development of the 1st computer that could store a program at Manchester University
- Turing committed suicide in the 50's

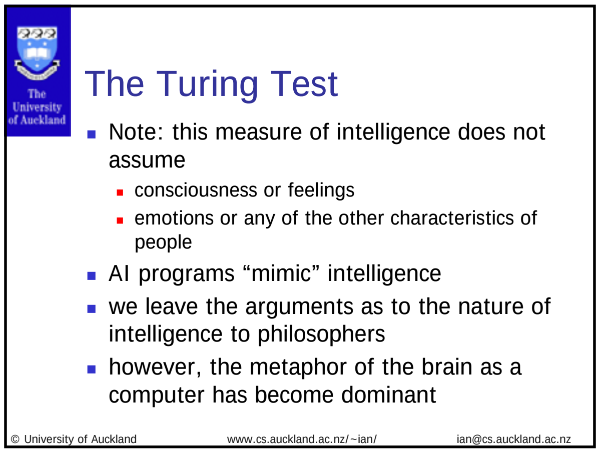
www.alanturing.net/

The Turing Test

- Can you decide which answers come from the person and which from the computer?

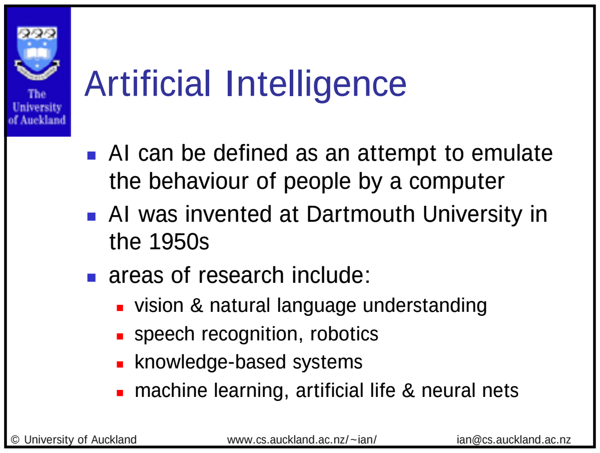


www.loebner.net/Prizef/loebner-prize.html



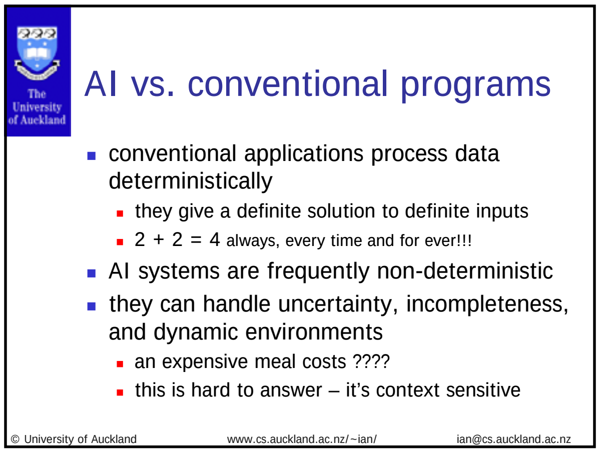
The Turing Test

- Note: this measure of intelligence does not assume
  - consciousness or feelings
  - emotions or any of the other characteristics of people
- AI programs “mimic” intelligence
- we leave the arguments as to the nature of intelligence to philosophers
- however, the metaphor of the brain as a computer has become dominant



Artificial Intelligence

- AI can be defined as an attempt to emulate the behaviour of people by a computer
- AI was invented at Dartmouth University in the 1950s
- areas of research include:
  - vision & natural language understanding
  - speech recognition, robotics
  - knowledge-based systems
  - machine learning, artificial life & neural nets



AI vs. conventional programs

- conventional applications process data deterministically
  - they give a definite solution to definite inputs
  - 2 + 2 = 4 always, every time and for ever!!!
- AI systems are frequently non-deterministic
- they can handle uncertainty, incompleteness, and dynamic environments
  - an expensive meal costs ????
  - this is hard to answer – it’s context sensitive

Symbol Systems

- AI programs reduce problems to symbols
- these symbols can be manipulated
- the manipulation of these symbols can seem intelligent
- the computer does not “know” what the symbols mean

Representing Problems as Symbols

A farmer has a problem, he has to cross a river by boat taking with him his dog, goose and a sack of corn. The boat is small and can only hold one item with the farmer.
He can’t leave the dog alone with the goose - the dog will eat the goose. He can’t leave the goose alone with the corn - the goose will eat the corn.

What is the order in which the farmer transfers his property across the river?

A Symbolic Representation

- Dog = d
- Goose = g
- Corn = c

symbols

- At the start of the problem all are on the left back of the river = L(d,g,c)
- The right bank is empty = R()

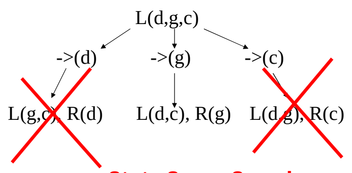
states

- row dog to right bank = ->(d)
- row corn to left bank = <-(c)

operators

A Symbolic Representation

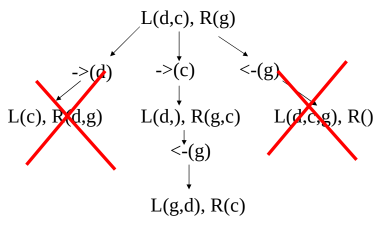
- Starting State = L(d,g,c), R()
- Goal State = L(), R(d,g,c)



State Space Search

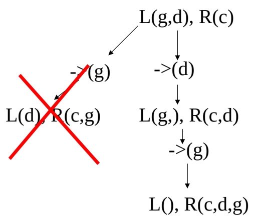
A Symbolic Representation

- Starting State = L(d,c), R(g)
- Goal State = L(), R(d,g,c)



A Symbolic Representation

- Starting State = L(g,d), R(c)
- Goal State = L(), R(d,g,c)



A Symbolic Representation

- Starting State = L(g,d,c), R()
- Goal State = L(), R(d,g,c)
- ->(g), ->(c), <-(g), ->(d), ->(g)

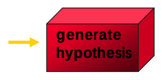
- Many AI programs use State Space Search
- This will be covered in detail later

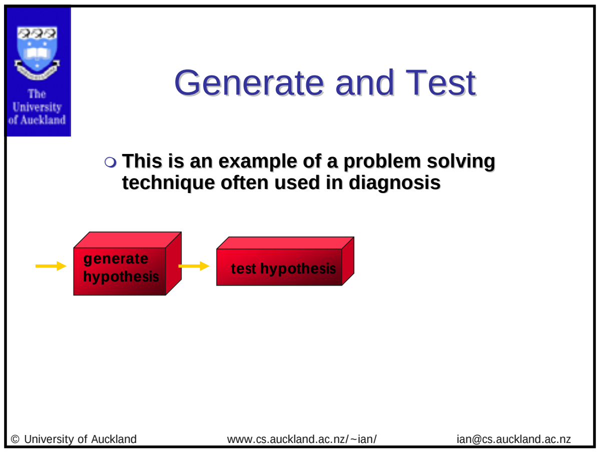
General Purpose Problem Solver (GPS)

- in 1963 Newell and Simon attempted to build a program that could solve problems like people
- the program did not contain knowledge about the world
- instead it attempted to generalise problem solving methods

Generate and Test

- **This is an example of a problem solving technique often used in diagnosis**





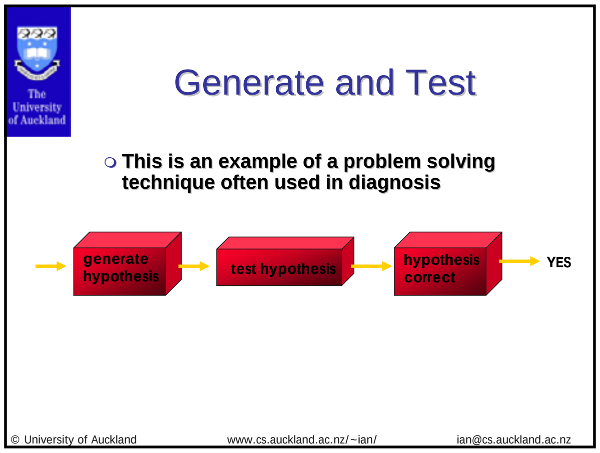
The University of Auckland

Generate and Test

- This is an example of a problem solving technique often used in diagnosis

generate hypothesis
test hypothesis

© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz



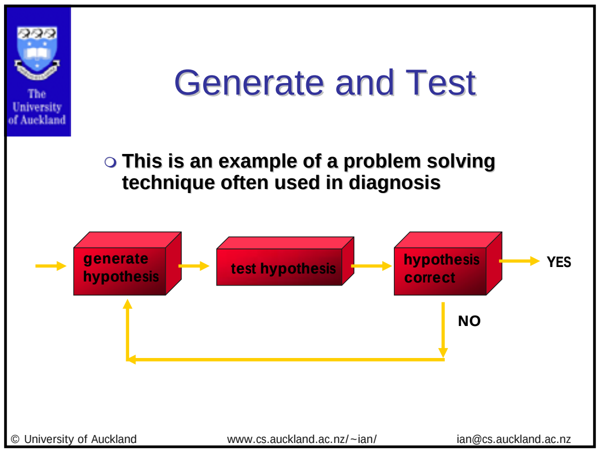
The University of Auckland

Generate and Test

- This is an example of a problem solving technique often used in diagnosis

generate hypothesis
test hypothesis
hypothesis correct
YES

© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz



The University of Auckland

Generate and Test

- This is an example of a problem solving technique often used in diagnosis

generate hypothesis
test hypothesis
hypothesis correct
YES
NO

© University of Auckland www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

GPS

- GPS was moderatly successful
- it could solve logical expressions
- and mathematical theorems
- To 1st year undergraduate level
- but not "*real world*" problems

Expert Systems

- It was realised that to solve problems you need knowledge about the problem area
  - Doctors need medical knowledge
- the knowledge must be stored as symbols that a program can manipulate to solve problems
- perhaps using problem solving methods such as generate and test

Expert Systems

- in the mid 70's several pioneering ES were built in the US
  - MYCIN - diagnosed infectious diseases of the blood
  - DENDRAL - analysed mass spectroscopy results
  - PROSPECTOR - analysed geological survey data to find mineral deposits
  - R1 - configured DEC VAX computers

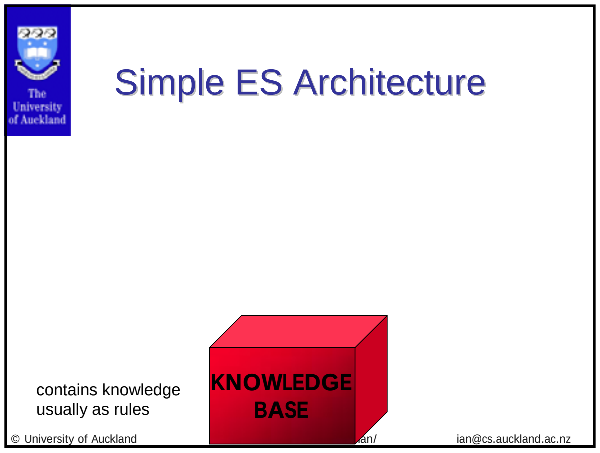
Expert Systems

- Expert System = Knowledge-Based System
- systems that embody expert knowledge in such a form that they can offer seemingly intelligent advice or decisions
- Require expert knowledge
- Need knowledge engineering
- This is dealt with later

Expert Systems

- communicate with users through a one-to-one dialogue
- justify *why* a question is being asked
- detect inconsistency in users' answers
- explain *how* a conclusion was reached
- separate knowledge about a problem from the control of the system

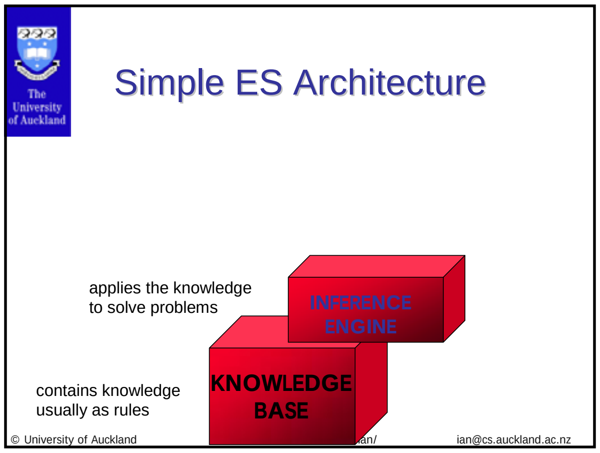
Simple ES Architecture



Simple ES Architecture

contains knowledge usually as rules

KNOWLEDGE BASE



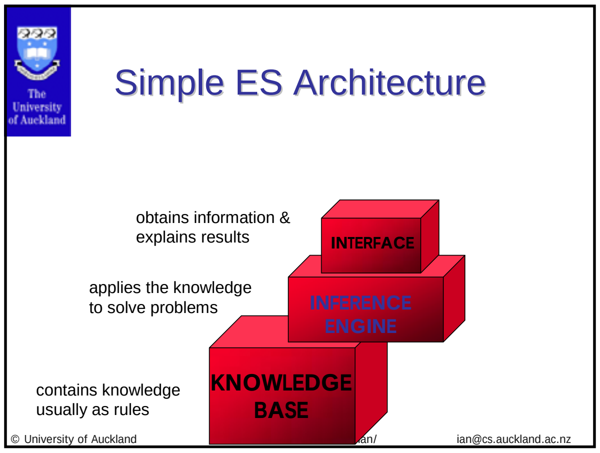
Simple ES Architecture

applies the knowledge to solve problems

INFERENCE ENGINE

contains knowledge usually as rules

KNOWLEDGE BASE



Simple ES Architecture

obtains information & explains results

INTERFACE

applies the knowledge to solve problems

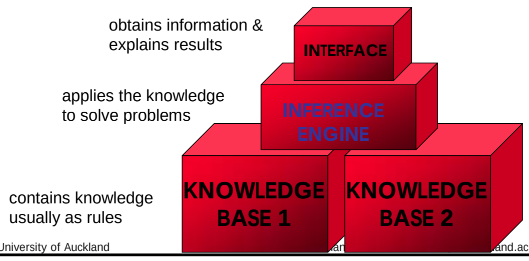
INFERENCE ENGINE

contains knowledge usually as rules

KNOWLEDGE BASE

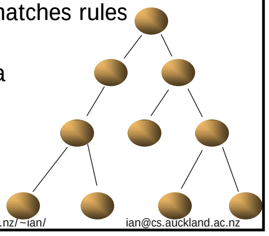
Simple ES Architecture

an interface and inference engine can use many different knowledge-bases = **expert system shell**


obtains information & explains results
INTERFACE
applies the knowledge to solve problems
INFERENCE ENGINE
contains knowledge usually as rules
KNOWLEDGE BASE 1
KNOWLEDGE BASE 2

Rule-Based Systems

- knowledge can be expressed as rules
- problems can be solved using rules
- rules are stored in the knowledge base
- the inference engine matches rules against data
- and can infer new data

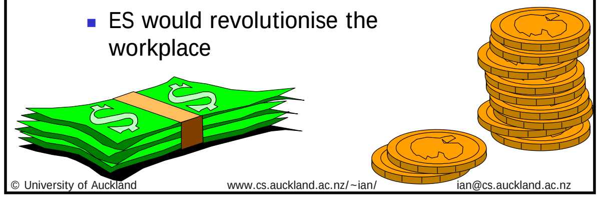


Rule-Based Systems

- Rule-based ES worked!!!
- they were simple
- they were relatively easy to program
- they mimicked how experts worked
- they could explain how they reached a conclusion
- they could be used for commercial benefit
- this was the BREAKTHROUGH AI needed

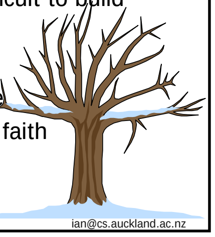
The Hype

- in the late 70’s AI gurus started claiming that ES would become a multi-million dollar business
- ES would operate in every industry
- ES would revolutionise the workplace



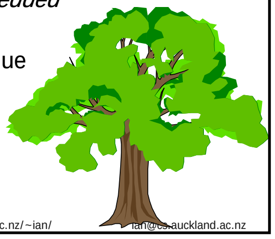
The AI Winter

- Large companies invested in AI and caught a chill
- ES were expensive and difficult to build
- ES were hard to maintain
- people didn’t like them
- few ES lived up to the hype
- companies lost money and faith



The AI Spring

- AI techniques have entered the main stream of IT (e.g. business-rules, objects & agents)
- AI has become “*embedded*”
- it is now just another programming technique
- AI makes money



The AI Spring

- Processor power is now very cheap
- The A-B route finding algorithm in SatNav systems (A* algorithm)
- Was invented in.....

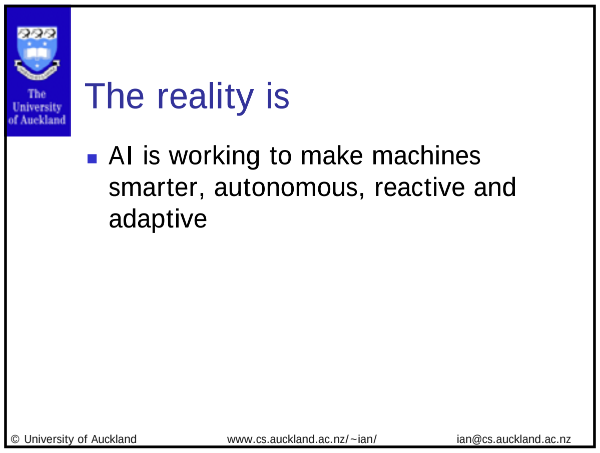
1968

The AI Spring

- Lots of well known AI techniques are now usable because of cheap processing power
- Called *brute force*
- AI is popular again

The AI Spring

- Consider Google http://labs.google.com/
- They want to employ people with skills in:
  - artificial intelligence
  - data mining
  - genetic algorithms
  - information retrieval
  - machine learning
  - natural language processing
  - robotics



The reality is

- AI is working to make machines smarter, autonomous, reactive and adaptive